

SUBTREE PERFECTNESS, BACKWARD INDUCTION, AND NORMAL-EXTENSIVE FORM EQUIVALENCE FOR SINGLE AGENT SEQUENTIAL DECISION MAKING UNDER ARBITRARY CHOICE FUNCTIONS

NATHAN HUNTLEY AND MATTHIAS C. M. TROFFAES

ABSTRACT. We revisit and reinterpret Selten’s concept of subgame perfectness in the context of single agent normal form sequential decision making, which leads us to the concept of subtree perfectness. Thereby, we extend Hammond’s characterization of extensive form consequentialist consistent behaviour norms to the normal form and to arbitrary choice functions under very few assumptions. In particular, we do not need to assume probabilities on any event or utilities on any reward. We show that subtree perfectness is equivalent to normal-extensive form equivalence, and is sufficient, but, perhaps surprisingly, not necessary, for backward induction to work.

1. INTRODUCTION

In a single agent sequential decision problem, at any stage, one has two ways of looking at its solution: the problem can be considered either in its simplest form—discarding any past stages, or as part of a much larger problem—possibly considering choices and events that did not actually obtain. A reasonable requirement is that, at any stage, the solution is independent of the larger problem it is embedded in. In this paper, we call this requirement *subtree perfectness*.

Selten [23] introduced a similar idea for multi-agent extensive form games, called *subgame perfectness*. In such games, players decide sequentially, and a player’s *behaviour strategy* specifies a decision at each point where he must choose. An *equilibrium point* is a strategy for each player, such that no player would change their strategy if they knew the strategy of all others.

A *subgame* is a part of a game that is again a game. For example, if a game consists of four rounds, then after having played two rounds, one can consider the remaining two rounds as a separate game. An equilibrium point of the full game of course implies a strategy for each player in the subgame simply by restricting the strategy of each player in the full game to only that subgame. If, for every subgame, the restriction of the equilibrium point of the full game to that subgame also yields an equilibrium point of that subgame, then the equilibrium point of the full game is called *subgame perfect*.

Date: 16 September, 2011.

Key words and phrases. choice function; subtree perfectness; sequential; normal form; extensive form; decision tree; backward induction; consequentialist.

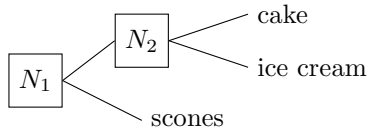


FIGURE 1. Two-stage problem.

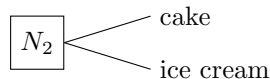


FIGURE 2. Second stage.

Selten showed that for games with perfect recall, (perfect¹) equilibrium points are subgame perfect [23, p. 39, Thm. 2], that is, they are independent of any larger game in which they could be embedded.

As mentioned, in this paper we investigate single agent sequential decision making modelled by decision trees. Although such problems differ in many ways from extensive form games, subtree perfectness is clearly analogous to subgame perfectness, as the following example shows.

Consider the decision problem in Fig. 1. In the first stage, the subject chooses between taking scones, or proceeding to the second stage. In the second stage, the subject chooses between cake, or ice cream. Suppose the subject prefers to reject scones and to choose ice cream at the second stage. This strategy induces a substrategy in the subtree for the second stage: choose ice cream over cake.

But, as with multi-agent games, we can instead consider the subtree for the second stage separately, as in Fig. 2. If, in this smaller tree, the subject prefers ice cream, then his solution is *subtree perfect*: his solution for the full tree induces a strategy in the subtree, and this strategy coincides with his solution for the subtree. If the subject states a difference preference (either no preference, or clear preference for cake), then his solution lacks subtree perfectness. So, subtree perfectness essentially means that the optimal induced strategies in a subtree do not depend on the full tree in which the subtree is embedded.

Our goal is to determine the conditions under which a theory of choice is subtree perfect. We consider a very large class of theories of choice, namely any that can be represented by conditional choice functions on gambles: we merely assume that for any set of gambles (functions from the possibility space Ω to a set of rewards \mathcal{R} ; these generalize random variables, or horse lotteries), and any conditioning event, the subject can give a non-empty subset of gambles that he considers optimal. Gambles that are non-optimal would never be selected, and the subject is unable to express further preference between the optimal ones. Maximizing expected utility is a simple example of such a choice function.

General choice functions, however, need neither probability nor utility—not even a preorder. Consequently, we must take care to distinguish clearly between the *normal form* and the *extensive form* of a decision problem (or game). While these forms are equivalent when maximizing expected utility [20, 1.3.4], their equivalence breaks down when expected utility is abandoned, as shown by for instance Seidenfeld [21], Machina [16], and Jaffray [9], among others.

Further, the standard definitions of normal and extensive form are not compatible with many choice functions. The usual extensive form solution for a decision

¹Without going into much detail, a *perfect equilibrium point* is one which is stable under small perturbations [23, p. 38].

tree, as given by Raiffa and Schlaifer [20, 1.2.1], involves using backward induction to replace subtrees with their maximum expected utility. Clearly, this definition is too restrictive for general choice functions, particularly those who do not correspond to a total preorder. Therefore, instead, we use the terms normal and extensive form in a far more general sense, while retaining their core features: the normal form specifies all actions in all eventualities in advance, whereas the extensive form specifies actions locally at each decision node.

Our main results start out from the traditional normal form method of listing all strategies, finding their corresponding gambles, applying a choice function, and listing all strategies that induce optimal gambles. In doing so, we assume act-state independence—that is, choice functions do not depend on the decision. Dropping act-state independence is beyond the scope of this paper. Under this assumption, we find three necessary and sufficient conditions for a choice function to induce a subtree perfect normal form solution. Further, we show that a normal form solution induced by a choice function satisfying these conditions will always have an equivalent extensive form representation. Interestingly, this does not hold for subtree perfect normal form solutions in general, only those induced by a choice function.

Our results are very similar to those of Hammond [6], although it may appear that Hammond’s goal is quite different from ours. Hammond considers extensive form solutions only, calls them *behaviour norms* [6, p. 28], and defines a notion of *consistency* which corresponds exactly to what we call subtree perfectness for extensive form solutions. Two trees are called *strategically equivalent* [16, p. 1636] if their normal form decisions (or, strategies) induce the same set of gambles. Hammond calls a behaviour norm *consequentialist* if it preserves strategic equivalence, that is, if strategically equivalent trees also have strategically equivalent behaviour norms. Hammond shows that a consistent and consequentialist behaviour norm implies a choice function on gambles, and then investigates the properties of this choice function. Although our goal, to find necessary and sufficient conditions on *arbitrary* choice functions under which they induce subtree perfect *normal* form solutions, is apparently different, Hammond’s approach effectively amounts to the same thing, but starting from extensive form, and only for a particular class of choice functions.

Even more so, because, as we will show, subtree perfectness implies normal-extensive form equivalence, unsurprisingly, Hammond’s necessary and sufficient conditions on a choice function for it to be implied by a consistent and consequentialist behaviour norm, are very similar to our necessary and sufficient conditions for subtree perfectness. Hammond’s first condition, that the choice function must induce a total preorder, is identical to our Property 2. His second, the independence axiom, is not present in our work because it only arises if probabilities can be assigned to certain chance arcs, an assumption we do not make. His third, a type of sure-thing principle, is very similar to our Property 3, where the difference is again based on whether or not probabilities are admitted. Finally, our Property 1 is not present in Hammond’s account, because of a small difference in the definition of decision trees.

Since our study is closely related to Hammond’s work, and makes use of many similar ideas, we shall throughout note where our definitions and properties coincide with those of Hammond, and also discuss the implications of the differences in

approach. It turns out that the only significant difference is dependent upon exactly how consequentialism is defined. Hammond writes a description of consequentialism that turns out to be essentially identical to our approach, but his mathematical definition that he uses in all proofs is actually a weaker version, with which one cannot prove results quite as strong as ours. In Theorem 37, we unify our work with Hammond’s informal but strong definition.

Subtree perfectness also relates to Machina’s [16, p. 1627] concept of *separability over mutually exclusive events*, which essentially boils down to subtree perfectness for probability trees. Although Machina defines separability using probabilities and a total preorder, it can be easily generalized for arbitrary choice functions and more general gambles that do not involve probabilities. All such possible generalizations seem to be equivalent to, or implied by, our Property 3.

A different concept called *separability* is defined by McClennen [17, p. 122]. Under the assumption of *dynamic consistency* [17, p. 120], this form of separability proves to be equivalent to subtree perfectness. The condition of dynamic consistency is implicitly assumed in our solutions. Although once again McClennen’s decision trees are somewhat different from ours, his Theorems 8.1 and 8.2 are very similar to our Lemma 21, and indeed the decision trees used in the proofs are almost identical.

Subtree perfectness is strongly linked to backward induction methods for solving decision trees. Indeed, under a total preorder, failures of backward induction are often used to illustrate violations of subtree perfectness [11, 16]. We find that subtree perfectness is sufficient, but, perhaps surprisingly, *not* necessary, for backward induction to work. In particular, a total preorder is not required. Further, backward induction implies a weaker form of subtree perfectness, in which every strategy must induce an optimal substrategy in every subtree, but the set of optimal strategies is not required to induce the set of all optimal substrategies in every subtree.

The paper is structured as follows: Section 2 explains decision trees and introduces notation. Section 3 provides a careful definition of normal and extensive form solutions, and introduces the concept of gambles to more easily work with normal form solutions. Section 4 introduces choice functions and their relationship with normal form solutions. Section 5 defines subtree perfectness and contains the principal results. Section 6 explores when a backward induction method can be used to find the normal form solution induced by a choice function. Section 7 examines the relationships subtree perfectness has with backward induction, extensive form equivalence, and the ideas of Hammond, McClennen, and Machina.

2. DECISION TREES

In this section, we explain the basic ideas behind decision trees and notation which will be used throughout the rest of the paper. For more information about decision trees, we refer to the literature [14, 3].

2.1. Definition and Example. A decision tree consists of a rooted tree [5, p. 92, Sec. 3.2] of decision nodes, chance nodes, and reward leaves, growing from left to right. The left hand side corresponds to what happens first, and the right hand side to what happens last. The payoffs for each sequence of decisions and events are at the end.

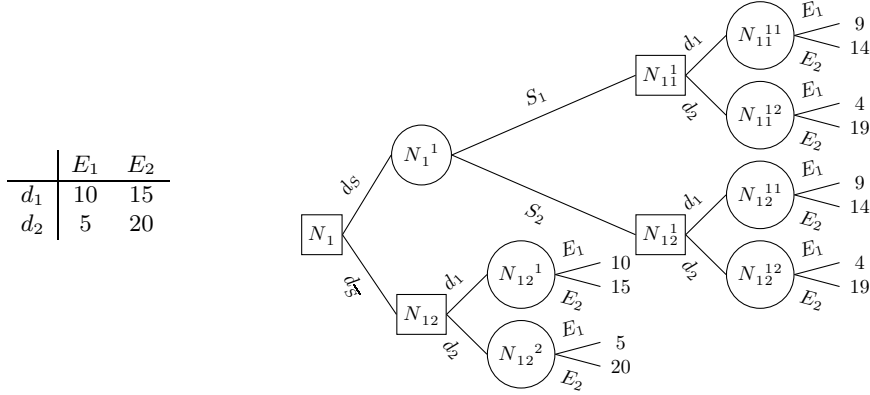


FIGURE 3. Payoff table and decision tree for walking in the lake district.

Consider the following example. Tomorrow, a subject is going for a walk in the lake district. It may rain (E_1), or not (E_2). The subject can either take a waterproof (d_1), or not (d_2), and can also choose to buy today's newspaper to learn tomorrow's weather forecast (d_S), or not ($d_{\bar{S}}$). Suppose that the forecast can have either two outcomes: predicting rain (S_1), or not (S_2). The utility of each combination, if the subject does not buy the newspaper, is summarized in Figure 3 (left). If the subject buys the newspaper, then one utile is subtracted.

Figure 3 also depicts the decision tree. Squares are decision nodes, and circles are chance nodes. From each node, branches emerge. For decision nodes, these are decisions; for chance nodes, these are events. For each chance node, the events form a partition of the possibility space, contrary to Hammond [6, p. 31], where events at chance nodes form a partition of the set of all states of nature still possible upon reaching the chance node. This leads to several technical differences, that will be seen in Definition 8, Property 1, and Property 7. Furthermore, our chance nodes are called *natural nodes* by Hammond. Hammond's chance nodes involve probabilities and do not appear in our decision trees.

In our example, the subject must first decide whether to buy the newspaper or not, hence we start with a decision node. If he buys the newspaper (d_S), then he learns about tomorrow's forecast. Thus, the chance node following d_S has two branches, forecasting rain (S_1), or no rain (S_2). Next, when leaving for the trip, he can either bring his waterproof (d_1) or not (d_2), hence the decision node following S_1 . During the walk, a chance node yields rain (E_1) or not (E_2).

So, each path in a decision tree amounts to a particular sequence of decisions and events. The payoffs resulting from each such sequence are put at the end.

2.2. Notation. Before elaborating how decision trees can be solved, we introduce a convenient mathematical notation for them. Decision trees can be seen as combinations of smaller decision trees: for instance, in the lake district example, one could draw the subtree corresponding to buying the newspaper, and also draw the subtree corresponding to making an immediate decision. The decision tree for the full problem is then formed by joining these two subtrees at a decision node.

Hence, we can represent a decision tree as follows. Let T_1, \dots, T_n be decision trees and E_1, \dots, E_n be a partition of the possibility space. If T is formed by

combining the trees at a decision node, we write

$$T = \bigsqcup_{i=1}^n T_i.$$

If T is formed by combining the trees at a chance node, with subtree T_i being connected by event E_i , we write

$$T = \bigodot_{i=1}^n E_i T_i.$$

For instance, for the tree of Fig. 3, we write

$$(S_1(T_1 \sqcup T_2) \odot S_2(T_1 \sqcup T_2)) \sqcup (U_1 \sqcup U_2)$$

with

$$T_1 = E_1 9 \odot E_2 14 \quad U_1 = E_1 10 \odot E_2 15 \quad T_2 = E_1 4 \odot E_2 19 \quad U_2 = E_1 5 \odot E_2 20$$

In certain circumstances it will be necessary to consider decision trees formed by adding a decision node in front of a tree T , that is, a decision tree whose root is a decision node with one option. Such a tree shall be denoted by $\sqcup T$.

In this paper we shall often be considering subtrees of larger trees. For subtrees, we need to know the events that were observed in the past. Two subtrees with the same configuration of nodes and arcs may have different preceding events, and should be treated differently. Therefore we associate with every decision tree T an event $\text{ev}(T)$: the intersection of all the events on chance arcs that have preceded T . Hammond [6, p. 27] denotes these events by $S(n)$.

Definition 1. A subtree of a tree T obtained by removal of all non-descendants of a particular node N is called the subtree of T at N and is denoted by $\text{st}_N(T)$.

Subtrees are called *continuation trees* by Hammond [6].

3. SOLVING DECISION TREES

This paper deals with more general solutions of decision trees than are usually considered. Consequently, the usual definitions of extensive and normal forms, such as in Raiffa and Schlaifer [20], are insufficient. Therefore, we first carefully define normal and extensive form solutions.

3.1. Extensive and Normal Form Solutions. The usual definition of extensive form is based on backward induction and expected utility. At each ultimate decision node, we calculate the expected utility of each option, and choose a maximal arc, replacing that node by its maximum expected utility. The penultimate decision nodes have now become the ultimate ones, so we can repeat this process, until the root node.

If options are not assigned values, then this type of backward induction cannot be used. Therefore, we abandon the link with backward induction, and instead focus on another property of extensive form: the decision arc to follow only needs to be specified when the subject actually reaches the decision node.

An *extensive form solution* of a decision tree removes from each decision node some (possibly none), but not all, of the decision arcs. So, an extensive form solution is a subtree of the original decision tree, where at each decision node a non-empty subset of arcs is retained. For instance, in the lake district example,

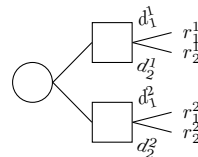
one of the extensive form solutions is: do not buy the newspaper, and then either take the waterproof or not. An extensive form solution can be used as follows: the subject, upon reaching a decision node, chooses one of the arcs in the extensive form solution, and follows it. The subject only needs to decide which arc to follow at a decision node when reaching that node.

In contrast, normal form solutions specify all future decisions at the start, so no further decisions need to be taken as time progresses. Classically, normal form solutions involve listing all possible combinations of actions and then choosing one to maximize expected utility. This generalizes as follows.

First, an extensive form solution with just one arc out of each decision node, is called a *normal form decision*. For instance, in the example, one of the normal form decisions is: buy the newspaper, and take the waterproof if the newspaper predicts rain, but do not take the waterproof otherwise. We denote the set of all normal form decisions for a decision tree T by $\text{nfd}(T)$. Normal form decisions are also called *strategies*, *pure strategies*, *plans*, and *policies*.

A *normal form solution* of a decision tree T is then simply a subset of $\text{nfd}(T)$. The interpretation of this subset is that the subject picks one of the normal form decisions of the normal form solution, and then acts accordingly.

It is natural to ask whether solutions of one form can be meaningfully transformed into solutions of the other. This question is addressed in Section 7.2. For now, note only that there are usually more normal form solutions than there are extensive form solutions. For example, consider the tree below, and suppose a normal form solution contains two normal form decisions, namely $d_1^1 d_2^2$ and $d_2^1 d_1^2$. Any attempt to find a corresponding extensive form solution will have to include all four decision arcs, but that would also correspond to the normal form solution where all four normal form decisions are present. Therefore there is no one-to-one correspondence between extensive and normal form solutions.



These two forms of solution are not the only possibilities. For instance, McClellenn [17] considers a form of solution where the subject must give a set of optimal normal form decisions *at every node in the tree*. The interpretation is apparently that the subject chooses a plan every time he reaches a node. We shall call such a solution a *dynamic normal form solution*. Our normal form solutions are special cases of dynamic normal form solutions, where the plan at any node is simply the restriction, to the node in question, of the plan at the root node (McClellenn calls this a *dynamically consistent solution*).

3.2. Extensive and Normal Form Operators. An *extensive form operator* is a function which maps each decision tree to an extensive form solution of that decision tree. The method by which decision arcs are removed is not specified, and in particular, need not be related to backward induction. Hammond [6, p. 28] calls these operators *behaviour norms*.

A *normal form operator* maps each decision tree to a normal form solution of that tree. Again, the method by which this happens is not part of our definition.

These operators are usually interpreted as describing optimality.

Clearly, the classical extensive and normal form interpretations are, respectively, extensive and normal form operators. Classical backward induction deletes any arc with non-maximal expectation, and results in a tree with some arcs deleted: an

	ω_1	ω_2	ω_3	ω_4
$E_1 9 \oplus E_2 14$	9	9	14	14
$S_1 (E_1 9 \oplus E_2 14) \oplus S_2 (E_1 4 \oplus E_2 19)$	9	4	14	19

TABLE 1. Example of normal form gambles.

extensive form solution. The classical normal form finds the set of normal form decisions with maximal expectation: a normal form solution.

3.3. Gambles. To express normal form decisions and solutions efficiently, we first introduce some definitions and notation. Let Ω be the *possibility space*: the set of all possible states of the world. We only consider finite possibility spaces. Elements of Ω are called typically denoted by ω . Subsets of Ω are called *events*. Let \mathcal{R} be a set of rewards. For our results we need *not* assume $\mathcal{R} = \mathbb{R}$.

A *gamble* is a function $X: \Omega \rightarrow \mathcal{R}$, and is interpreted as an uncertain reward: once $\omega \in \Omega$ is observed, X yields $X(\omega)$. Probabilities over Ω are *not* needed.

3.4. Normal Form Gambles. Once a normal form decision is chosen, the reward is determined entirely by the events that obtain. So, every normal form decision has a corresponding gamble, which we call a *normal form gamble*. The set of all normal form gambles associated with a decision tree T is denoted by $\text{gamb}(T)$.

Using Fig. 3, we explain how to find normal form gambles. First consider the subtree at N_{11}^1 , which has normal form decisions d_1 and d_2 . The former gives reward 9 utiles if $\omega \in E_1$ and 14 utiles if $\omega \in E_2$, and so yields the gamble

$$(1) \quad E_1 9 \oplus E_2 14.$$

The \oplus operator combines partial maps defined on disjoint domains (i.e. the partial map $E_1 9$ defined on E_1 , and the partial map $E_2 14$ defined on E_2).

Now consider the subtree with root at N_1^1 , and in particular the normal form decision ‘ d_1 if S_1 and d_2 if S_2 ’. This gives reward 9 if $\omega \in S_1 \cap E_1$, reward 14 if $\omega \in S_1 \cap E_2$, and so on. The corresponding gamble is

$$(S_1 \cap E_1) 9 \oplus (S_1 \cap E_2) 14 \oplus (S_2 \cap E_1) 4 \oplus (S_2 \cap E_2) 19,$$

or briefly, if we omit ‘ \cap ’ and employ distributivity,

$$(2) \quad S_1 (E_1 9 \oplus E_2 14) \oplus S_2 (E_1 4 \oplus E_2 19),$$

where multiplication with an event is now understood to correspond to restriction, i.e., 9 is a constant map on Ω , $E_1 9$ is a constant map restricted to E_1 , and $S_1 (E_1 9)$ is obtained from $E_1 9$ by further restriction to $E_1 \cap S_1$. For illustration, we tabulate the values of some normal form gambles in Table 1, where $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$, $E_1 = \{\omega_1, \omega_2\}$, and $S_1 = \{\omega_1, \omega_3\}$.

Observe that the gamble in Eq. (2) incorporates the gamble in Eq. (1). This observation allows a very convenient recursive definition of the gamb operator.

Definition 2. For any events E_1, \dots, E_n which form a partition, and any finite family of sets of gambles $\mathcal{X}_1, \dots, \mathcal{X}_n$, we define the following set of gambles:

$$(3) \quad \bigoplus_{i=1}^n E_i \mathcal{X}_i = \left\{ \bigoplus_{i=1}^n E_i X_i : X_i \in \mathcal{X}_i \right\}$$

Definition 3. With any decision tree T , we associate a set of gambles $\text{gamb}(T)$, recursively defined through:

- If a tree T consists of only a leaf with reward $r \in \mathcal{R}$, then

$$(4a) \quad \text{gamb}(T) = \{r\}.$$

- If a tree T has a chance node as root, that is, $T = \odot_{i=1}^n E_i T_i$, then

$$(4b) \quad \text{gamb}\left(\odot_{i=1}^n E_i T_i\right) = \bigoplus_{i=1}^n E_i \text{gamb}(T_i).$$

- If a tree T has a decision node as root, that is, if $T = \sqcup_{i=1}^n T_i$, then

$$(4c) \quad \text{gamb}\left(\sqcup_{i=1}^n T_i\right) = \bigcup_{i=1}^n \text{gamb}(T_i).$$

It is easy to show that this definition gives indeed the required set,

$$(5) \quad \text{gamb}(T) = \bigcup_{U \in \text{nfd}(T)} \text{gamb}(U).$$

Hammond uses $F(T, n)$ for $\text{gamb}(\text{st}_N(T))$. McClennen uses $G(T)$ for $\text{gamb}(T)$.

Multiple decision trees can model the same problem. This suggests the following definition (see for instance [12, 16]):

Definition 4. Two decision trees T_1 and T_2 are called strategically equivalent if $\text{gamb}(T_1) = \text{gamb}(T_2)$.

Hammond [6, p. 38] uses the term *consequentially equivalent*.

4. NORMAL FORM SOLUTIONS FOR DECISION TREES

4.1. Choice Functions and Optimality. We defined a normal form solution as a subset of all normal form decisions. Ideally one would like to identify a single best normal form decision, but this is not always possible. The subject might, however, still be able to eliminate some normal form decisions that he would never consider choosing, leaving a subset of normal form decisions. We say that the subject considers these as *optimal*.

In classical decision theory, each normal form decision induces a random real-valued gain, and is considered optimal if its expected gain is maximized. As another example, suppose that a set \mathcal{M} of plausible probability distributions are specified. Then the subject might consider optimal all normal form decisions whose expected gain is maximal under at least one distribution in \mathcal{M} .

So, often, optimal decisions are determined by comparison of gambles. We follow this common approach, since normal form decisions have corresponding gambles, and gambles are easier to work with. We therefore suppose that the subject can determine an optimal subset of any set of gambles, conditional upon an event A (corresponding to $\text{ev}(T)$ of the tree T in question):

Definition 5. A choice function opt maps, for any non-empty event A , each non-empty finite set \mathcal{X} of gambles to a non-empty subset of this set:

$$\emptyset \neq \text{opt}(\mathcal{X}|A) \subseteq \mathcal{X}.$$

Note that common uses of choice functions in social choice theory, such as by Sen [24, p. 63, ll. 19–21] do not consider conditioning, and define choice functions for arbitrary sets of options (not for gambles only).

4.2. Normal Form Operator Induced by a Choice Function. Now, given a choice function opt , we naturally arrive at a normal form operator norm_{opt} , simply by applying opt on the set of all gambles associated with the tree T and then finding the corresponding set of normal form decisions.

Definition 6. *Given any choice function opt , and any decision tree T with $\text{ev}(T) \neq \emptyset$, we define*

$$\text{norm}_{\text{opt}}(T) = \{U \in \text{nfd}(T) : \text{gamb}(U) \subseteq \text{opt}(\text{gamb}(T)|\text{ev}(T))\}.$$

Of course, since the U are normal form decisions, $\text{gamb}(U)$ is always a singleton in this definition. In particular, the following important equality holds,

$$(6) \quad \text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{opt}(\text{gamb}(T)|\text{ev}(T)).$$

It follows immediately that norm_{opt} respects strategic equivalence:

Theorem 7. *If T_1 and T_2 are strategically equivalent and $\text{ev}(T_1) = \text{ev}(T_2) \neq \emptyset$, then $\text{gamb}(\text{norm}_{\text{opt}}(T_1)) = \text{gamb}(\text{norm}_{\text{opt}}(T_2))$.*

This is of course an attractive property, as there are always many strategically equivalent trees representing the same problem: Theorem 7 guarantees that all equivalent representations yield the same solution.

When studying subtree perfectness, we consider norm_{opt} for arbitrary subtrees. To ensure that norm_{opt} can be applied on each of these, we need:

Definition 8. *A decision tree T is called consistent if for every node N of T ,*

$$\text{ev}(\text{st}_N(T)) \neq \emptyset.$$

Clearly, if a decision tree T is consistent, then for any node N in T , $\text{st}_N(T)$ is also consistent. We study only consistent decision trees because we consider $\text{norm}_{\text{opt}}(\text{st}_N(T))$ for any node N in T , which is impossible when $\text{ev}(\text{st}_N(T)) = \emptyset$.

Usually, one does not consider events which conflict with preceding events, hence consistency is satisfied. However, due to an oversight, some branch of a chance node might represent an event that cannot occur: such tree can always be made consistent by removing those nodes whose conditioning event is empty.

Not all sets of gambles can be represented by a consistent decision tree:

Definition 9. *Let A be any non-empty event, and let \mathcal{X} be a non-empty finite set of gambles. Then the following conditions are equivalent; if any (hence all) of them are satisfied, we say that \mathcal{X} is A -consistent.*

- (A) *There is a consistent decision tree T with $\text{ev}(T) = A$ and $\text{gamb}(T) = \mathcal{X}$.*
- (B) *For every $r \in \mathcal{R}$ and every $X \in \mathcal{X}$ such that $X^{-1}(r) \neq \emptyset$, it holds that $X^{-1}(r) \cap A \neq \emptyset$.*

We will also say that a gamble X is A -consistent whenever $\{X\}$ is A -consistent.

5. SUBTREE PERFECTNESS

We now define subtree perfectness, for both types of operator, and find necessary and sufficient conditions on opt for norm_{opt} to be subtree perfect.

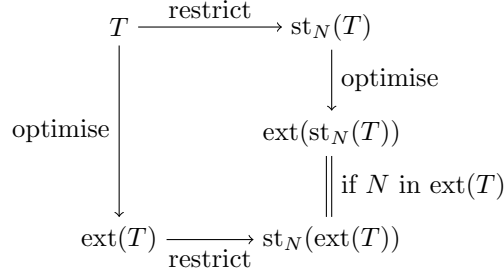


FIGURE 4. For a subtree perfect extensive form operator, optimization and restriction commute.

5.1. Example and Definition. First, we illustrate subtree perfectness by an example. Suppose we apply an extensive form operator to the tree T in Fig. 3. This operator will delete some (possibly none) of the decision arcs at $N = N_{11}^1$. If the operator would delete the same arcs at N regardless of the larger tree in which $\text{st}_N(T)$ is embedded, then the operator is subtree perfect. If the operator does not have this property (for instance, if the solution of T after N were to depend on consequences of $d_{\overline{S}}$ or S_2), then it fails subtree perfectness. Hammond [6, p. 34] calls subtree perfectness for behaviour norms (extensive form operators) *consistency*.

The definition for subtree perfectness for a normal form operator requires the following extension to Definition 1.

Definition 10. If \mathcal{T} is a set of decision trees and N a node, then

$$\text{st}_N(\mathcal{T}) = \{\text{st}_N(T) : T \in \mathcal{T} \text{ and } N \text{ in } T\}.$$

Definition 11. An extensive form operator ext is called subtree perfect if for every consistent decision tree T and every node N such that N is in $\text{ext}(T)$,

$$\text{st}_N(\text{ext}(T)) = \text{ext}(\text{st}_N(T)).$$

A normal form operator norm is called subtree perfect if for every consistent decision tree T and every node N which is in at least one element of $\text{norm}(T)$,

$$\text{st}_N(\text{norm}(T)) = \text{norm}(\text{st}_N(T)).$$

In other words, for a subtree perfect operator, it does not matter whether we first restrict to a subtree and then optimize, or first optimize and only then restrict to the subtree: subtree perfectness means that optimization and restriction commute, as in Fig. 4.

The extensive form operator ext_P corresponding to the usual backward induction using expected utility is well known to be subtree perfect, provided probabilities at chance nodes are non-zero. Also, the usual normal form operator norm_P corresponding to maximizing expected utility over all normal form decisions is subtree perfect, because ext_P is equivalent to norm_P .

Not all choice functions are subtree perfect:

Example 12. Let T be the decision tree in Fig. 5, where X , Y , and Z are its normal form gambles. Under point-wise dominance, X and Y are incomparable, as are Y and Z . Hence, $\text{norm}(\text{st}_N(T))$ is $\{X, Y\}$ (where we conveniently identified normal form decisions with their normal form gambles). But $\text{norm}(T) = \text{opt}(\{X, Y, Z\}) =$

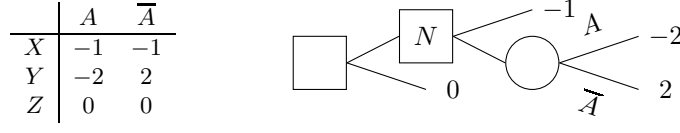


FIGURE 5. Decision tree for Example 12.

$\{Y, Z\}$ as clearly Z dominates X . Restricting this solution to $\text{st}_N(T)$ gives the normal form solution $\{Y\}$. Concluding,

$$\{X, Y\} = \text{norm}(\text{st}_N(T)) \neq \text{st}_N(\text{norm}(T)) = \{Y\}$$

and therefore the normal form operator induced by opt lacks subtree perfectness.

5.2. Necessary and Sufficient Conditions. In this section, we work extensively with normal form solutions, which are sets of trees. Therefore, we extend gamb , \odot , and \sqcup , to sets of trees:

Definition 13. For any set of decision trees \mathcal{T} , $\text{gamb}(\mathcal{T}) = \bigcup_{T \in \mathcal{T}} \text{gamb}(T)$.

Definition 14. For any sets of consistent decision trees $\mathcal{T}_1, \dots, \mathcal{T}_n$, and any partition E_1, \dots, E_n , let

$$\begin{aligned} \odot_{i=1}^n E_i \mathcal{T}_i &= \{\odot_{i=1}^n E_i T_i : T_i \in \mathcal{T}_i\}, \\ \sqcup_{i=1}^n \mathcal{T}_i &= \{\sqcup_{i=1}^n T_i : T_i \in \mathcal{T}_i\}. \end{aligned}$$

For sets of trees, the gamb operator keeps working as expected:

$$(7) \quad \text{gamb}(\odot_{i=1}^n E_i \mathcal{T}_i) = \bigoplus_{i=1}^n E_i \text{gamb}(\mathcal{T}_i),$$

$$(8) \quad \text{gamb}(\sqcup_{i=1}^n \mathcal{T}_i) = \bigcup_{i=1}^n \text{gamb}(\mathcal{T}_i).$$

One should also observe that $\text{gamb}(T) = \text{gamb}(\text{nfd}(T))$.

The next three properties turn out to be necessary and sufficient for subtree perfectness of normal form operators induced by a choice function.

Property 1 (Conditioning Property). Let A be a non-empty event, and let \mathcal{X} be a non-empty finite A -consistent set of gambles, with $\{X, Y\} \subseteq \mathcal{X}$ such that $AX = AY$. If $X \in \text{opt}(\mathcal{X}|A)$, then $Y \in \text{opt}(\mathcal{X}|A)$.

This property is not found in the accounts of Hammond and McClennen, because their decision trees are different from ours. If their methods were adapted to our decision trees, then the conditioning property would appear in their work too. The property is more of a technical detail than an important point, since it states that, if two gambles are equal on an event, one gamble cannot be preferred to the other given that event. Such a property would be common and desirable for a choice function, regardless of its implications for subtree perfectness.

Property 2 (Intersection property). For any event $A \neq \emptyset$ and non-empty finite A -consistent sets of gambles \mathcal{X} and \mathcal{Y} such that $\mathcal{Y} \subseteq \mathcal{X}$ and $\text{opt}(\mathcal{X}|A) \cap \mathcal{Y} \neq \emptyset$,

$$\text{opt}(\mathcal{Y}|A) = \text{opt}(\mathcal{X}|A) \cap \mathcal{Y}.$$

The intersection property is precisely Arrow's (C4) [2, p. 123].

For the next property, we use the following extension of the \oplus notation: if A is a non-trivial event (non-empty and not Ω), then

$$A\mathcal{X} \oplus \overline{A}Z = \{AX \oplus \overline{A}Z : X \in \mathcal{X}\}.$$

Property 3 (Mixture property). *For any events A and B such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, any $\overline{A} \cap B$ -consistent gamble Z , and any non-empty finite $A \cap B$ -consistent set of gambles \mathcal{X} ,*

$$\text{opt}(A\mathcal{X} \oplus \overline{A}Z|B) = A\text{opt}(\mathcal{X}|A \cap B) \oplus \overline{A}Z.$$

This property is a form of the well known independence principle (see for example [17, p. 44]). It has strongest similarities to McClennen's independence for choice [17, p. 57], and Arrow's conditional preference [1, p. 257].

Property 2 has a many equivalent formulations. The following three give interesting alternative interpretations, and are useful in some of the proofs.

Property 4 (Strong path independence). *For any non-empty event A and any non-empty finite A -consistent sets of gambles $\mathcal{X}_1, \dots, \mathcal{X}_n$, there is a non-empty $\mathcal{I} \subseteq \{1, \dots, n\}$ such that*

$$\text{opt}\left(\bigcup_{i=1}^n \mathcal{X}_i \middle| A\right) = \bigcup_{i \in \mathcal{I}} \text{opt}(\mathcal{X}_i|A)$$

Property 5 (Very strong path independence). *For any non-empty event A and any non-empty finite A -consistent sets of gambles $\mathcal{X}_1, \dots, \mathcal{X}_n$,*

$$\text{opt}\left(\bigcup_{i=1}^n \mathcal{X}_i \middle| A\right) = \bigcup_{\substack{i=1 \\ \mathcal{X}_i \cap \text{opt}(\bigcup_{i=1}^n \mathcal{X}_i|A) \neq \emptyset}}^n \text{opt}(\mathcal{X}_i|A)$$

Property 6 (Total preorder). *For every event $A \neq \emptyset$, there is a total preorder \succeq_A on A -consistent gambles such that for every non-empty finite set of A -consistent gambles \mathcal{X} ,*

$$\text{opt}(\mathcal{X}|A) = \{X \in \mathcal{X} : (\forall Y \in \mathcal{X})(X \succeq_A Y)\}$$

The total preorder property essentially boils down to Arrow's (C5), which is also called the *weak axiom of revealed preference* [2, p. 123]. More equivalents are given by Arrow [2, (C1)], Houthakker [7, p. 163], and Ville [25, p. 123].

Lemma 15. *Properties 2, 4, 5 and 6 are equivalent.*

Proof. Property 6 \implies Property 5 \implies Property 4 \implies Property 2. Immediate. Property 2 \implies Property 6. See Arrow's (C4) \iff (C5) [2, p. 124, Thm. 1]. \square

To show that Properties 1, 2 and 3 are necessary and sufficient for subtree perfectness of norm_{opt} , we require several lemmas. The proofs are long but mostly tedious and straightforward, and so are omitted.

For a decision tree T , $\text{ch}(T)$ is the set of child nodes of the root node of T .

Lemma 16. *Let norm be any normal form operator. Let T be a consistent decision tree. If,*

(i) *for all nodes $K \in \text{ch}(T)$ such that K is in at least one element of $\text{norm}(T)$,*

$$\text{st}_K(\text{norm}(T)) = \text{norm}(\text{st}_K(T)),$$

(ii) and, for all nodes $K \in \text{ch}(T)$, and all nodes $L \in \text{st}_K(T)$ such that L is in at least one element of $\text{norm}(\text{st}_K(T))$,

$$\text{st}_L(\text{norm}(\text{st}_K(T))) = \text{norm}(\text{st}_L(\text{st}_K(T))),$$

then, for all nodes N in T such that N is in at least one element of $\text{norm}(T)$,

$$\text{st}_N(\text{norm}(T)) = \text{norm}(\text{st}_N(T)).$$

Lemma 17. Let A_1, \dots, A_n be a finite partition of Ω , and let B be an event such that $A_i \cap B \neq \emptyset$ for all i . Let $\mathcal{X}_1, \dots, \mathcal{X}_n$ be a finite family of non-empty finite sets of gambles, where \mathcal{X}_i is $A_i \cap B$ -consistent. If a choice function opt satisfies Properties 2 and 3, then

$$(9) \quad \text{opt} \left(\bigoplus_{i=1}^n A_i \mathcal{X}_i \middle| B \right) = \bigoplus_{i=1}^n A_i \text{opt}(\mathcal{X}_i | A_i \cap B).$$

Lemma 18. Consider a consistent decision tree T whose root is a decision node, so $T = \bigsqcup_{i=1}^n T_i$, and any choice function opt . For each tree T_i , let N_i be its root. Then, N_i is in at least one element of $\text{norm}_{\text{opt}}(T)$ if and only if

$$(10) \quad \text{gamb}(T_i) \cap \text{opt}(\text{gamb}(T) | \text{ev}(T)) \neq \emptyset.$$

Lemma 19. For any consistent decision tree $T = \odot_{i=1}^n E_i T_i$, and any choice function opt satisfying Property 1,

$$(11) \quad \text{gamb}(\text{norm}_{\text{opt}}(T)) = \bigoplus_{i=1}^n E_i \text{gamb}(\text{norm}_{\text{opt}}(T_i))$$

implies

$$\text{norm}_{\text{opt}}(T) = \bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i).$$

Lemma 20. For any consistent decision tree $T = \bigsqcup_{i=1}^n T_i$ and any choice function opt satisfying Property 2,

$$(12) \quad \text{gamb}(\text{norm}_{\text{opt}}(T)) = \bigcup_{i \in \mathcal{I}} \text{gamb}(\text{norm}_{\text{opt}}(T_i))$$

implies

$$\text{norm}_{\text{opt}}(T) = \text{nfd} \left(\bigsqcup_{i \in \mathcal{I}} \text{norm}_{\text{opt}}(T_i) \right),$$

where $\mathcal{I} = \{i \in \{1, \dots, n\} : \text{gamb}(T_i) \cap \text{opt}(\text{gamb}(T) | \text{ev}(T)) \neq \emptyset\}$.

The next lemma shows necessity of Properties 1, 2, and 3 for subtree perfectness. Interestingly, the proof only involves the two decision trees in Figure 6.

Lemma 21. If norm_{opt} is subtree perfect, then opt satisfies Properties 1, 2, and 3.

We are now ready to establish that Properties 1, 2, and 3 are also sufficient for subtree perfectness.

Theorem 22 (Subtree perfectness theorem). A normal form operator norm_{opt} is subtree perfect if and only if opt satisfies Properties 1, 2 and 3.

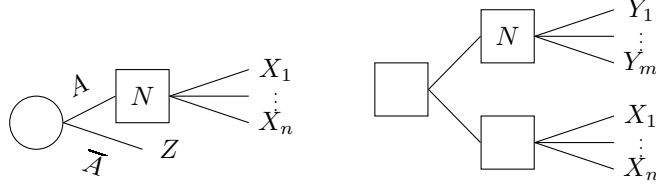


FIGURE 6. Decision trees for Lemma 21.

Proof. “only if”. See Lemma 21.

“if”. We proceed by structural induction on all possible arguments of norm_{opt} , that is, on all consistent decision trees. In the base step, we prove the implication for trees consisting of only a single node. In the induction step, we prove that if the implication holds for the subtrees at every child of the root node, then the implication also holds for the whole tree.

First, if the decision tree T has only a single node (a reward, and no further children) then subtree perfectness is trivially satisfied.

Next, suppose that the consistent decision tree T has multiple nodes. Let $\{N_1, \dots, N_n\} = \text{ch}(T)$ be the children of the root node of T , and let $T_i = \text{st}_{N_i}(T)$. The induction hypothesis is: subtree perfectness holds for all subtrees at every child of the root node, that is, for all T_i . More precisely, for all $i \in \{1, \dots, n\}$, and all nodes $L \in T_i$ which are in at least one element of $\text{norm}_{\text{opt}}(T_i)$

$$\text{st}_L(\text{norm}_{\text{opt}}(T_i)) = \text{norm}_{\text{opt}}(\text{st}_L(T_i)).$$

We must show that

$$\text{st}_N(\text{norm}_{\text{opt}}(T)) = \text{norm}_{\text{opt}}(\text{st}_N(T))$$

for all nodes N in T such that N is in at least one element of $\text{norm}_{\text{opt}}(T)$. By Lemma 16, and the induction hypothesis, it suffices to prove the above equality only for $N \in \text{ch}(T)$, that is, it suffices to show that

$$(13) \quad \text{st}_{N_i}(\text{norm}_{\text{opt}}(T)) = \text{norm}_{\text{opt}}(T_i)$$

for each $i \in \{1, \dots, n\}$ such that N_i is in at least one element of $\text{norm}_{\text{opt}}(T)$.

If T has a chance node as its root, that is, $T = \bigodot_{i=1}^n E_i T_i$, then all N_i are actually in every element of $\text{norm}_{\text{opt}}(T)$, so we must simply establish Eq. (13) for all $i \in \{1, \dots, n\}$. Observe that, if we can establish

$$(14) \quad \text{norm}_{\text{opt}}(T) = \bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i),$$

then Eq. (13) follows immediately. Indeed, by Eq. (6),

$$\text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{opt}(\text{gamb}(T) | \text{ev}(T))$$

and by the definition of the gamb operator, Eq. (4b) in particular,

$$= \text{opt} \left(\bigoplus_{i=1}^n E_i \text{gamb}(T_i) \middle| \text{ev}(T) \right)$$

and so by Lemma 17,

$$= \bigoplus_{i=1}^n E_i \text{ opt}(\text{gamb}(T_i) | \text{ev}(T) \cap E_i)$$

so, since $\text{ev}(T) \cap E_i = \text{ev}(T_i)$, and again by Eq. (6),

$$= \bigoplus_{i=1}^n E_i \text{ gamb}(\text{norm}_{\text{opt}}(T_i))$$

Whence, Eq. (14) follows by Lemma 19.

Finally, assume that T has a decision node as its root, that is, $T = \bigsqcup_{i=1}^n T_i$. Let \mathcal{I} be the subset of $\{1, \dots, n\}$ such that $i \in \mathcal{I}$ if and only if N_i is in at least one element of $\text{norm}_{\text{opt}}(T)$. We must establish Eq. (13) for all $i \in \mathcal{I}$. Equivalently, we must show that

$$(15) \quad \text{norm}_{\text{opt}}(T) = \text{nfd} \left(\bigsqcup_{i \in \mathcal{I}} \text{norm}_{\text{opt}}(T_i) \right).$$

Indeed, by Eq. (6),

$$\text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{opt}(\text{gamb}(T) | \text{ev}(T))$$

and by the definition of the gamb operator, Eq. (4c) in particular,

$$= \text{opt} \left(\bigcup_{i=1}^n \text{gamb}(T_i) \middle| \text{ev}(T) \right)$$

and so by Property 5,

$$= \bigcup_{i \in \mathcal{I}^*} \text{opt}(\text{gamb}(T_i) | \text{ev}(T)),$$

where $\mathcal{I}^* = \{i \in \{1, \dots, n\} : \text{gamb}(T_i) \cap \text{opt}(\text{gamb}(T) | \text{ev}(T)) \neq \emptyset\}$, and so because $\text{ev}(T) = \text{ev}(T_i)$, and again by Eq. (6),

$$= \bigcup_{i \in \mathcal{I}^*} \text{gamb}(\text{norm}_{\text{opt}}(T_i)).$$

Hence, the conditions of Lemma 20 are satisfied, and $\mathcal{I}^* = \mathcal{I}$ by Lemma 18, so Eq. (15) is established. \square

6. BACKWARD INDUCTION

Although Selten's [23] definition of subgame perfectness does not explicitly refer to backward induction, one of its aims is unmistakably to identify when a game's equilibrium point can be found by backward induction. In fact, when maximizing expected utility, as in Selten's work, there seems little reason to distinguish between subtree perfectness and backward induction because the optimal decisions are essentially unique: multiple optimal decisions will have the same expectation, so it suffices to consider only a single representative.

However, if our concept of choice has no reference to expectation, multiple optimal decisions are not necessarily equivalent in any sense. Consequently, backward induction does not need to be tied to subtree perfectness.

Therefore, elsewhere [8], we suggested the following backward induction method, which generalizes classical backward induction to arbitrary choice functions, and which is useful when $\text{gamb}(T)$ is very large and applying opt in one go is not feasible. We first extend norm_{opt} to act on sets of decision trees.

Definition 23. *Given a choice function opt and any set \mathcal{T} of consistent decision trees, where $\text{ev}(T) = A$ for all $T \in \mathcal{T}$,*

$$\text{norm}_{\text{opt}}(\mathcal{T}) = \{U \in \text{nfd}(\mathcal{T}) : \text{gamb}(U) \subseteq \text{opt}(\text{gamb}(\mathcal{T})|A)\}.$$

The goal of our backward induction algorithm is to reach a normal form solution of T by finding normal form solutions of subtrees of T , and using these solutions to remove some elements of $\text{gamb}(T)$ before applying opt :

Definition 24. *The normal form operator back_{opt} is defined for any consistent decision tree T through:*

- *If T consists of only a leaf with reward $r \in \mathcal{R}$, then*

$$(16a) \quad \text{back}_{\text{opt}}(T) = \{T\}.$$

- *If T has a chance node as root, that is, $T = \bigodot_{i=1}^n E_i T_i$, then*

$$(16b) \quad \text{back}_{\text{opt}}\left(\bigodot_{i=1}^n E_i T_i\right) = \text{norm}_{\text{opt}}\left(\bigodot_{i=1}^n E_i \text{back}_{\text{opt}}(T_i)\right)$$

- *If T has a decision node as root, that is, if $T = \bigsqcup_{i=1}^n T_i$, then*

$$(16c) \quad \text{back}_{\text{opt}}\left(\bigsqcup_{i=1}^n T_i\right) = \text{norm}_{\text{opt}}\left(\bigsqcup_{i=1}^n \text{back}_{\text{opt}}(T_i)\right).$$

This algorithm is almost identical to that suggested by Kikuti et al. [10]: they apply opt only at decision nodes, whereas we apply opt at all type of node.

It is instructive to compare the definition of the gamb operator (Definition 3) with Definition 24. The main difference is that back_{opt} inserts norm_{opt} at every stage, to remove as many normal form decisions as possible, early on.

If $\text{back}_{\text{opt}} = \text{norm}_{\text{opt}}$, we can use the former as an efficient way of calculating the latter. Of course, this only works if a normal form gamble that is non-optimal in a subtree at a node cannot be part of an optimal gamble in the full tree. It is well known that choice functions exist for which this property does not hold: for examples, see LaValle and Wapman [11], Jaffray [9], and Seidenfeld [22]. In such cases, there exist trees such that $\text{back}_{\text{opt}}(T) \neq \text{norm}_{\text{opt}}(T)$.

The following four properties are necessary and sufficient for back_{opt} to coincide with norm_{opt} .

Property 7 (Backward conditioning property). *Let A and B be events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, and let \mathcal{X} be a non-empty finite $A \cap B$ -consistent set of gambles, with $\{X, Y\} \subseteq \mathcal{X}$ such that $AX = AY$. Then $X \in \text{opt}(\mathcal{X}|A \cap B)$ implies $Y \in \text{opt}(\mathcal{X}|A \cap B)$ whenever there is a non-empty finite $\overline{A} \cap B$ -consistent set of gambles \mathcal{Z} such that, for at least one $Z \in \mathcal{Z}$,*

$$AX \oplus \overline{A}Z \in \text{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B).$$

This is clearly just a minor relaxation of Property 1.

Property 8 (Insensitivity of optimality to the omission of non-optimal elements). *For any event $A \neq \emptyset$, and any non-empty finite A -consistent sets of gambles \mathcal{X} and \mathcal{Y} ,*

$$\text{opt}(\mathcal{X}|A) \subseteq \mathcal{Y} \subseteq \mathcal{X} \Rightarrow \text{opt}(\mathcal{Y}|A) = \text{opt}(\mathcal{X}|A).$$

If opt satisfies this property, then removing non-optimal elements from a set does not affect whether or not each of the remaining elements is optimal. The property is called ‘insensitivity to the omission of non-optimal elements’ by De Cooman and Troffaes [4], and ‘property ϵ ’ by Sen [24] who attributes this designation to Douglas Blair.

Property 9 (Preservation of non-optimality under the addition of elements). *For any event $A \neq \emptyset$, and any non-empty finite A -consistent sets of gambles \mathcal{X} and \mathcal{Y} ,*

$$\mathcal{Y} \subseteq \mathcal{X} \Rightarrow \text{opt}(\mathcal{Y}|A) \supseteq \text{opt}(\mathcal{X}|A) \cap \mathcal{Y}.$$

This is ‘property α ’ in Sen [24], Axiom 7 in Luce and Raiffa [15, p. 288], and ‘independence of irrelevant alternatives’ in Radner and Marschak [19].

Property 10 (Backward mixture property). *For any events A and B such that $B \cap A \neq \emptyset$ and $B \cap \bar{A} \neq \emptyset$, any $B \cap \bar{A}$ -consistent gamble Z , and any non-empty finite $B \cap A$ -consistent set of gambles \mathcal{X} ,*

$$\text{opt}(A\mathcal{X} \oplus \bar{A}Z|B) \subseteq A\text{opt}(\mathcal{X}|A \cap B) \oplus \bar{A}Z.$$

This is an “inclusion-only” version of Property 3. We do not need the full property because backward induction moves from right to left.

The proof of the following theorem is, up to obvious details, identical to the proof of Theorem 22, and is therefore left as an amusing exercise to the reader.

Theorem 25 (Backward induction theorem). *A normal form operator norm_{opt} satisfies backward induction (that is, $\text{back}_{\text{opt}}(T) = \text{norm}_{\text{opt}}(T)$ for every consistent decision tree T) if and only if opt satisfies Properties 7, 8, 9, and 10.*

If $\text{back}_{\text{opt}}(T) = \text{norm}_{\text{opt}}(T)$ for any consistent T , then, at decision nodes, $\text{back}_{\text{opt}}(T)$ coincides with the method of Kikuti et al. [10].

7. DISCUSSION

7.1. Relationship Between Subtree Perfectness and Backward Induction.

Obviously, Property 1 implies Property 7, and Property 3 implies Property 10. Also, it is easily shown that:

Lemma 26. *Property 2 implies Properties 8 and 9.*

Corollary 27. *If norm_{opt} is subtree perfect, then $\text{norm}_{\text{opt}} = \text{back}_{\text{opt}}$.*

Subtree perfectness is, however, not necessary for backward induction. For example, point-wise dominance satisfies Properties 1, 8, 9, and 10, but as we saw in Example 12, it lacks subtree perfectness.

Backward induction does imply a weaker form of subtree perfectness. Suppose $\text{back}_{\text{opt}}(T) = \text{norm}_{\text{opt}}(T)$ for all consistent decision trees. By definition of back_{opt} , for any node N that is in at least one element of $\text{back}_{\text{opt}}(T)$ we have

$$\text{st}_N(\text{norm}_{\text{opt}}(T)) = \text{st}_N(\text{back}_{\text{opt}}(T)) \subseteq \text{back}_{\text{opt}}(\text{st}_N(T)) = \text{norm}_{\text{opt}}(\text{st}_N(T)).$$

Why can this be seen as a type of subtree perfectness? A subgame perfect equilibrium point is one that induces an equilibrium point in all subgames. A subtree perfect normal form operator is one that induces its normal form solution in all subtrees. Theorem 25 implies *every* optimal normal form *decision* induces an optimal normal form decision in any subtree. So although we do not have subtree perfectness of *solutions* we do have subtree perfectness of *decisions*.

Definition 28. *norm is subtree perfect for normal form decisions if for every N in T such that N is in at least one element of $\text{norm}(T)$,*

$$\text{st}_N(\text{norm}(T)) \subseteq \text{norm}(\text{st}_N(T)).$$

Obviously, backward induction implies subtree perfectness for normal form decisions. But, the opposite implication does not hold: Property 8 is not necessary for subtree perfectness for normal form decisions.

Theorem 29. *For any choice function opt , norm_{opt} is subtree perfect for normal form decisions if and only if opt satisfies Properties 7, 9, and 10.*

7.2. Normal Form and Extensive Form Equivalence. Extensive form solutions can be seen as more natural solutions to decision trees, because they represent the sequential nature of the decision making, whereas the normal form solutions remove the sequential aspects. It is common to ask whether there is an equivalence between certain normal form operators and extensive form operators. For our definitions of normal and extensive form, this is very different from the of “normal form/extensive form coincidence” of McClennen [17, p. 115]. In McClennen’s terms, the normal form and extensive form refer to the structure of the tree, and coincidence requires the solution of the two forms to be the same. In the case of norm_{opt} , this coincidence will always occur by definition.

The clearest equivalence arises when $\text{norm}(T) = \{\text{ext}(T)\}$, that is, when the extensive form solution is itself a normal form decision, and is also the only element of the normal form solution. Even when using choice functions corresponding to total preorders, this may not arise for every decision tree, so a more general concept of equivalence is required. We shall consider an extensive form operator and a normal form operator to be equivalent if, for every consistent decision tree T ,

$$\text{norm}(T) = \text{nfd}(\text{ext}(T)).$$

If such an equivalence holds, it is easy to move from the extensive form solution to the normal form solution, simply by finding all normal form decisions. It is also easy to move from the normal form to extensive form, as the following result shows.

Lemma 30. *Suppose norm and ext are equivalent. Then a node N is in $\text{ext}(T)$ if and only if N is in at least one element of $\text{norm}(T)$.*

If we can find equivalent normal form and extensive form operators, then either both are subtree perfect, or neither is.

Lemma 31. *Suppose that ext and norm are equivalent. Then, norm is subtree perfect if and only if ext is subtree perfect.*

With these results at hand, normal-extensive form equivalence is easily established, using structural induction as usual (the proof is left to the reader):

Theorem 32. *If a normal form operator norm_{opt} induced by a choice function opt is subtree perfect, then there exists an equivalent subtree perfect extensive form operator ext .*

Perhaps surprisingly, there are subtree perfect normal form operators (necessarily *not* induced by a choice function) which have no equivalent extensive form operator. Indeed, the proof of Theorem 32 relies on the following consequence of Lemmas 17 and 19:

$$\text{norm}_{\text{opt}}(\bigodot_{i=1}^n E_i T_i) = \bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i),$$

General subtree perfect normal form operators need not satisfy this. For instance, the example of Section 3.1 can yield a subtree perfect normal form operator which has no equivalent extensive form representation.

7.3. Related Work. As mentioned earlier, our results have strong links with the work of Hammond [6], Machina [16], and McClennen [17].

Machina [16] assumes probabilities at chance nodes and choice functions that correspond to total preorders. Under these assumptions, a necessary condition for subtree perfectness of norm_{opt} is that opt satisfies *separability over mutually exclusive events*. Consider n possible events with probabilities p_1, \dots, p_n , a gamble giving reward $r_i \in \mathcal{R}$ if event i obtains, and a second gamble that gives reward r_i if event i occurs for $i > 1$ and $r_* \in \mathcal{R}$ for $i = 1$. Separability says that the second gamble is preferred to the first if and only if r_* is preferred to r_1 .

There are various ways to adapt separability for our more general setting. For example, consider a non-trivial event A and gambles X and Y such that $\overline{AX} = \overline{AY}$, $AX = Ar_1$, and $AY = Ar_2$ for some rewards r_1 and $r_2 \in \mathcal{R}$. Separability could be: $\text{opt}(\{X, Y\}) = \{X\}$ if and only if $\text{opt}(\{r_1, r_2\}) = \{r_1\}$. Note that Property 3 implies this. Indeed, Property 3 can be seen as a strong form of separability, for it implies every reasonable generalization of separability.

As already noted, Hammond’s [6] results are slightly different from ours due to the definition of decision trees, and whether gambles involving probabilities are admitted. If such details are dealt with, then his results become very similar to ours. Using our terminology and notation, Hammond’s first defines:

Definition 33. *An extensive form operator ext is consistent if it is subtree perfect.*

Definition 34. *An extensive form operator ext is consequentialist if, for any decision trees T_1 and T_2 such that $\text{gamb}(T_1) = \text{gamb}(T_2)$ and $\text{ev}(T_1) = \text{ev}(T_2)$,*

$$\text{gamb}(\text{ext}(T_1)) = \text{gamb}(\text{ext}(T_2)).$$

Definition 35. *An extensive form operator ext is strongly consequentialist if it is consequentialist and, for any $U \in \text{nfd}(T)$ such that $\text{gamb}(U) \subseteq \text{gamb}(\text{ext}(T))$, $U \in \text{nfd}(\text{ext}(T))$.*

Note that Hammond does not use the term “strongly consequentialist”. He writes that consequentialism means that decisions should be valued by their gambles, which corresponds to strong consequentialism. Yet, his only mathematical definition of consequentialism seems to be identical to Definition 34. The difference is small, but important to link our results with his.

Hammond argues that consistent and (not necessarily strongly) consequentialist extensive form operators induce a choice function on gambles as follows.

Definition 36. For a consistent and consequentialist extensive form operator ext , define its corresponding choice function opt_{ext} by

$$\text{opt}_{\text{ext}}(\mathcal{X}|A) = \text{gamb}(\text{ext}(T)),$$

where T is any consistent decision tree with $\text{gamb}(T) = \mathcal{X}$ and $\text{ev}(T) = A$. Because ext is consistent and consequentialist, this choice function exists and does not depend on the choice of T .

With these definitions, we can prove a slightly stronger version of Hammond's results [6, Theorem 5.4, Theorem 6, Theorem 7, and Theorem 8].

Theorem 37. A choice function opt satisfies Properties 1, 2, and 3 if and only if there is a consistent and strongly consequentialist extensive form operator ext such that $\text{opt}_{\text{ext}} = \text{opt}$.

Proof. “if”. Follow the approach of Hammond [6, Theorem 5.4 and Theorem 7]. Note that these proofs require only consequentialism.

“only if”. Suppose opt satisfies Properties 1, 2, and 3. By Theorems 22 and 32, norm_{opt} is subtree perfect and has an equivalent subtree perfect extensive form operator ext . By definition of norm_{opt} , for any strategically equivalent trees T_1 and T_2 , $\text{gamb}(\text{norm}_{\text{opt}}(T_1)) = \text{gamb}(\text{norm}_{\text{opt}}(T_2))$, and so the same holds for ext . Hence, ext is consistent and consequentialist. By construction, ext is also strongly consequentialist, and obviously also $\text{opt}_{\text{ext}} = \text{opt}$. \square

It is easily seen that, for a particular choice function opt , there is exactly one consistent and strongly consequentialist extensive form operator that induces opt . Therefore, there is an equivalence between the consistent and *strongly* consequentialist extensive form operator inducing opt and the subtree perfect normal form operator induced by opt . This equivalence is *not* present in Hammond's account, since multiple consistent and (*not strongly*) consequentialist extensive form operators can induce the same choice function.

For example, if T has two normal form decisions inducing the same gamble X , $\text{nfd}(\text{ext}_1(T))$ could only include one, while $\text{nfd}(\text{ext}_2(T))$ includes both, without violating subtree perfectness. Moreover, $\text{gamb}(T)$ contains X so strategic equivalence is preserved as required. Hence, both operators can be consistent and consequentialist, however only at most one of these can be equivalent to norm_{opt} . Indeed, a consistent and consequentialist extensive form operator can *only* be equivalent to its corresponding norm_{opt} if it is *strongly* consequentialist.

As with Hammond, McClennen's [17] decision trees differ in that some chance nodes can have probabilities for events. Also, there seems to be no concept of conditioning in McClennen's account. As noted in Section 3, McClennen's dynamic normal form solutions are more general types of normal form solutions. Some of his results [17, Theorems 8.1 and 8.2] are similar to ours, and are based on three restrictions placed on his solutions.

Definition 38 (McClennen, [17, p. 120]). A dynamic normal form solution satisfies dynamic consistency if, for every node N in the tree, the restriction of the optimal set at the root node to N is exactly the optimal set at N .

Obviously, there is a one-to-one correspondence between dynamic normal form solutions satisfying dynamic consistency, and our normal form solutions.

Definition 39 (McClennen,[17, p. 114]). *A dynamic normal form solution satisfies plan reduction if, for every normal form decision in T that induces the same gamble, either all or none of them are optimal.*

By definition, norm_{opt} satisfies plan reduction.

Definition 40 (McClennen,[17, p. 122]). *A dynamic normal form solution satisfies separability if, for any tree T and any node N in T , the set of optimal plans at N is the same as the set of optimal plans of the separate tree $\text{st}_N(T)$.*

On its own, separability is not exactly subtree perfectness, but if dynamic consistency holds then the two properties are equivalent. McClennen’s two theorems can then be adapted into our setting as:

Theorem 41. *If a dynamic normal form solution satisfies plan reduction, dynamic consistency, and separability, then it coincides with norm_{opt} for a choice function opt satisfying Properties 1, 2, and 3.*

Proof. The proof is essentially identical to that of Lemma 21. □

8. CONCLUSION

We extended Selten’s idea of subtree perfectness to decision trees, for normal and extensive form solutions. Subtree perfectness for extensive form solutions is Hammond’s consistency condition. Subtree perfectness for normal form solutions is, under the assumption of dynamic consistency, McClennen’s separability. We found necessary and sufficient conditions for a choice function to induce a subtree perfect normal form operator. These turned out to be similar to, but stronger than, those for backward induction to work. So, even if a normal form operator lacks subtree perfectness, it may still be possible to find the normal form solution by backward induction.

While many choice functions satisfy Property 1, Properties 2 and 3 are perhaps more restrictive than one would like. Is violating subtree perfectness acceptable? We believe that subtree perfectness is a desirable property and one must think carefully before abandoning it. On the other hand, if one is attracted to the three properties for other reasons, then subtree perfectness gives them a strong justification, particularly since (at least for Properties 2 and 3) they are much more difficult to justify in a static setting. Attempts to justify violation of subtree perfectness, without violating dynamic consistency, have however been made, for example by Machina [16] and McClennen [17, 9.6].

We recovered the well-known fact that subtree perfectness requires total pre-ordering. Many choice functions suggested in the literature, such as maximality [26, Sec. 3.9] and E-admissibility [13], violate total preordering, and hence fail subtree perfectness. Interestingly, we can easily establish that some of these (particularly, maximality and E-admissibility) still admit backward induction.

If one is committed to the idea of subtree perfectness but also wishes to use a choice function that fails Property 2, then the best solution may be to use an extensive form operator. One can easily define subtree perfect normal form solutions based on choice functions (but not directly induced as in Definition 6). These, however, can have unpleasant behaviour such as admitting pointwise dominated options. It is easier to avoid such behaviour with subtree perfect extensive form solutions, as proposed for instance by Seidenfeld [21], although Seidenfeld’s idea

will only work if the choice function can model complete ignorance. Roughly, this solution is found by backward induction, and when a decision node admits multiple optimal options, then it is treated as a chance node with complete ignorance about which of the optimal decisions is chosen.

We have seen that, when using choice functions on gambles, subtree perfectness is closely related to equivalence between extensive form and normal form operators, both using our approach of defining an operator based on a choice function, and Hammond’s approach of defining a choice function based on an operator. Interestingly, normal-extensive form equivalence need *not* hold for subtree perfect operators that are not induced by choice functions.

Although we have primarily investigated norm_{opt} in this paper, we do not argue that a normal form operator, and norm_{opt} in particular, gives the best solution to a decision tree. A normal form solution requires a policy for all eventualities to be specified and adhered to. The subject adheres to this policy only by his own resolution: he may of course have the ability to change his policy upon reaching a decision node [21]. One could therefore argue that a normal form solution is only acceptable for sequential problems when the subject does not get the chance to change his mind (for example, if he instructs, in advance, others to carry out the actions).

In practice, applying opt to the set of all normal form gambles may be difficult. Therefore, we defined a normal form operator which yields a normal form solution by means of backward induction, and which will, in many cases, be easier to apply than norm_{opt} . We found necessary and sufficient conditions on opt for our backward induction algorithm to yield exactly norm_{opt} . As mentioned, we found that total preordering is not necessary for backward induction to work: the set of choice functions that satisfy the backward induction properties are a subset of those that satisfy subtree perfectness for normal form solutions and a superset of those that satisfy subtree perfectness. Hence, it remains unclear whether backward induction has any justification beyond practicality.

ACKNOWLEDGEMENTS

The authors are indebted to Teddy Seidenfeld for suggesting the term ‘subtree perfectness’, and to Wlodek Rabinowicz for suggesting the link between separability and subtree perfectness. EPSRC supports the first author.

REFERENCES

- [1] K. Arrow. Exposition of the theory of choice under uncertainty. *Synthese*, 16(3–4):253–269, Dec 1966.
- [2] Kenneth J. Arrow. Rational choice functions and orderings. *Economica*, 26(102):121–127, May 1959.
- [3] Robert T. Clemen and Terence Reilly. *Making Hard Decisions*. Duxbury, 2001.
- [4] G. De Cooman and M. C. M. Troffaes. Dynamic programming for deterministic discrete-time systems with uncertain gain. *International Journal of Approximate Reasoning*, 39(2–3):257–278, Jun 2005.
- [5] Jonathan Gross and Jay Yellen. *Graph Theory and Its Applications*. CRC Press, London, 1999.
- [6] P. Hammond. Consequentialist foundations for expected utility. *Theory and Decision*, 25(1):25–78, Jul 1988.
- [7] H. S. Houthakker. Revealed preference and the utility function. *Economica*, 17(66):159–174, May 1950.

- [8] Nathan Huntley and Matthias C. M. Troffaes. An efficient normal form solution to decision trees with lower previsions. In Didier Dubois, M. Asunción Lubiano, Henri Prade, María Ángeles Gil, Przemysław Grzegorzewski, and Olgierd Hryniewicz, editors, *Soft Methods for Handling Variability and Imprecision*, Advances in Soft Computing, pages 419–426. Springer, Sep 2008.
- [9] J. Jaffray. Rational decision making with imprecise probabilities. In *1st International Symposium on Imprecise Probabilities and Their Applications*, 1999.
- [10] D. Kikuti, F. Cozman, and C.P. de Campos. Partially ordered preferences in decision trees: Computing strategies with imprecision in probabilities. In R. Brafman and U. Junker, editors, *IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, pages 118–123, 2005.
- [11] I. LaValle and K. Wapman. Rolling back decision trees requires the independence axiom! *Management Science*, 32(3):382–385, Mar 1986.
- [12] I. H. LaValle and P. C. Fishburn. Equivalent decision trees and their associated strategy sets. *Theory and Decision*, 23(1):37–63, 1987.
- [13] I. Levi. *The Enterprise of Knowledge*. MIT Press, London, 1980.
- [14] D. V. Lindley. *Making Decisions*. Wiley, London, 2nd edition, 1985.
- [15] R. D. Luce and H. Raiffa. *Games and Decisions: introduction and critical survey*. Wiley, 1957.
- [16] M. J. Machina. Dynamic consistency and non-expected utility models of choice under uncertainty. *Journal of Economic Literature*, 27(1622-1688), 1989.
- [17] E. F. McClennen. *Rationality and Dynamic Choice: Foundational Explorations*. Cambridge University Press, 1990.
- [18] C. R. Plott. Path independence, rationality, and social choice. *Econometrica*, 41(6):1075–1091, Nov 1973.
- [19] R. Radner and J. Marschak. Note on some proposed decision criteria. In R. M. Thrall, C. H. Coombs, and R. L. Davies, editors, *Decision Processes*, pages 61–68. John Wiley, 1954.
- [20] H. Raiffa and R. Schlaifer. *Applied Statistical Decision Theory*. Harvard University Press, 1961.
- [21] T. Seidenfeld. Decision theory without ‘independence’ or without ‘ordering’: What is the difference? *Economics and Philosophy*, 4:267–290, 1988.
- [22] T. Seidenfeld. A contrast between two decision rules for use with (convex) sets of probabilities: Γ -maximin versus E -admissibility. *Synthese*, 140:69–88, 2004.
- [23] R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4(1):25–55, Mar 1975.
- [24] A. K. Sen. Social choice theory: A re-examination. *Econometrica*, 45(1):53–89, 1977.
- [25] Jean Ville and P. K. Newman. The existence-conditions of a total utility function. *The Review of Economic Studies*, 19(2):123–128, 1951.
- [26] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.

APPENDIX A. PROOFS

A.1. Proof of Eq. (5).

Lemma 42. *For any decision tree T , $\text{gamb}(T) = \text{gamb}(\text{nfd}(T))$.*

Proof. We prove this by structural induction. In the base step, we prove the equality for trees comprising only one node. In the induction step, we prove that if the equality holds for the subtrees at every child of the root node of T , then the equality also holds for T .

If T consists of only a single node, namely a reward node, then $\text{nfd}(T) = \{T\}$ and the result holds trivially. Thus the base step is confirmed.

Suppose T has a chance node at the root, that is, $T = \bigodot_{i=1}^n E_i T_i$. Each element of $\text{nfd}(T)$ is of the form $\bigodot_{i=1}^n U_i$, where $U_i \in \text{nfd}(T_i)$. In other words, $\text{nfd}(T)$ is the set of all possible mixtures of the elements of $\text{nfd}(T_i)$, that is,

$$\text{nfd}(T) = \bigodot_{i=1}^n E_i \text{nfd}(T_i).$$

The induction hypothesis is $\text{gamb}(T_i) = \text{gamb}(\text{nfd}(T_i))$ for each i . We have

$$\begin{aligned} \text{gamb}(\text{nfd}(T)) &= \text{gamb}\left(\bigodot_{i=1}^n E_i \text{nfd}(T_i)\right) \\ &= \bigoplus_{i=1}^n E_i \text{gamb}(\text{nfd}(T_i)) \\ &= \bigoplus_{i=1}^n E_i \text{gamb}(T_i) \\ &= \text{gamb}(T). \end{aligned}$$

On the other hand, if T has a decision node as a root, that is, $T = \bigsqcup_{i=1}^n T_i$, then

$$\text{nfd}(T) = \left\{ \sqcup U : U \in \bigcup_{i=1}^n \text{nfd}(T_i) \right\},$$

and, since $\text{gamb}(\sqcup U) = \text{gamb}(U)$ for any U ,

$$\begin{aligned} \text{gamb}(\text{nfd}(T)) &= \left\{ \text{gamb}(\sqcup U) : U \in \bigcup_{i=1}^n \text{nfd}(T_i) \right\} \\ &= \left\{ \text{gamb}(U) : U \in \bigcup_{i=1}^n \text{nfd}(T_i) \right\} \\ &= \bigcup_{i=1}^n \text{gamb}(\text{nfd}(T_i)) \end{aligned}$$

and again, the induction hypothesis says that $\text{gamb}(T_i) = \text{gamb}(\text{nfd}(T_i))$ for each i , so

$$\begin{aligned} &= \bigcup_{i=1}^n \text{gamb}(T_i) \\ &= \text{gamb}(T). \end{aligned}$$

This completes the induction step. \square

A.2. Equivalence of Definition 9(A) and (B).

Proof of equivalence. (A) \implies (B). We prove the implication by structural induction on the tree T . In the base step, we prove that, for every \mathcal{X} and non-empty event A , the implication holds for consistent decision trees which consist of only a single node. In the induction step, we prove that if, for all \mathcal{X} and non-empty A , the implication holds for the subtrees at every child of the root node, then, for all \mathcal{X} and non-empty A , the implication also holds for the whole tree.

First, if T consists of only a single node, namely a reward node, then $\text{gamb}(T) = \{s\}$ for some $s \in \mathcal{R}$, so by assumption, $\mathcal{X} = \text{gamb}(T) = \{X\}$ where X is the gamble yielding a constant value s . Clearly, $X^{-1}(r) = \emptyset$ for all $r \neq s$ and $X^{-1}(s) = \Omega$, hence indeed $X^{-1}(r) \cap A = A \neq \emptyset$ whenever $X^{-1}(r) \neq \emptyset$, and this for every non-empty event A .

Next, suppose T has a chance node as its root, that is, $T = \odot_{i=1}^n E_i T_i$. By assumption, and by the definition of gamb (see Definition 3),

$$\mathcal{X} = \text{gamb}(T) = \bigoplus_{i=1}^n E_i \text{gamb}(T_i) = \bigoplus_{i=1}^n E_i \mathcal{X}_i$$

where \mathcal{X}_i is a shorthand notation for $\text{gamb}(T_i)$. Also, by assumption, $\text{ev}(T) = A$. By the induction hypothesis, we know that for every $r \in \mathcal{R}$ and every $X_i \in \mathcal{X}_i$ such that $X_i^{-1}(r) \neq \emptyset$, it holds that $X_i^{-1}(r) \cap A \cap E_i \neq \emptyset$ (indeed, $\text{ev}(T_i) = \text{ev}(T) \cap E_i = A \cap E_i$). Now we have all ingredients to prove the desired implication. Indeed, for any $X \in \mathcal{X}$, or equivalently, for any $X = \bigoplus_{i=1}^n E_i X_i$ with $X_i \in \mathcal{X}_i$,

$$(17) \quad X^{-1}(r) = \left(\bigoplus_{i=1}^n E_i X_i \right)^{-1} (r) = \bigcup_{i=1}^n X_i^{-1}(r) \cap E_i$$

Hence Eq. (17) implies that $X^{-1}(r) \neq \emptyset$ whenever $X_i^{-1}(r) \cap E_i \neq \emptyset$ for at least one $i \in \{1, \dots, n\}$, and in that case, it obviously follows that also $X_i^{-1}(r) \neq \emptyset$, which implies, as just shown, that $X_i^{-1}(r) \cap A \cap E_i \neq \emptyset$. But, then, again by Eq. (17), it must also hold that $X^{-1}(r) \cap A \neq \emptyset$.

Finally, suppose that the root of T is a decision node, that is, $T = \bigsqcup_{i=1}^n T_i$. By assumption, and by the definition of gamb (see Definition 3),

$$\mathcal{X} = \text{gamb}(T) = \bigcup_{i=1}^n \text{gamb}(T_i) = \bigcup_{i=1}^n \mathcal{X}_i$$

where \mathcal{X}_i is a shorthand notation for $\text{gamb}(T_i)$. Also, by assumption, $\text{ev}(T) = A$. By the induction hypothesis, we know that for every $r \in \mathcal{R}$ and every $X_i \in \mathcal{X}_i$ such that $X_i^{-1}(r) \neq \emptyset$, it holds that $X_i^{-1}(r) \cap A \neq \emptyset$ (indeed, $\text{ev}(T_i) = \text{ev}(T) = A$). But, for any $X \in \mathcal{X}$, it follows that $X = X_i$ for some $X_i \in \mathcal{X}_i$, so the desired implication follows immediately from the induction hypothesis.

(B) \implies (A). Suppose that for every $r \in \mathcal{R}$ and every $X \in \mathcal{X}$ such that $X^{-1}(r) \neq \emptyset$, it holds that $X^{-1}(r) \cap A \neq \emptyset$. Consider the decision tree

$$T = \bigsqcup_{X \in \mathcal{X}} \bigodot_{\substack{r \in \mathcal{R} \\ X^{-1}(r) \neq \emptyset}} X^{-1}(r) r$$

with $\text{ev}(T) = A$. Let $N(X)$ denote the chance node of T associated with X , and let $N(X, r)$ denote the reward node of T associated with X and r (of course $N(X, r)$ only exists for $X^{-1}(r) \neq \emptyset$, by definition of T).

Clearly, T is consistent, because $\text{ev}(\text{st}_{N(X)}(T)) = \text{ev}(T) = A \neq \emptyset$ and $\text{ev}(\text{st}_{N(X, r)}(T)) = \text{ev}(\text{st}_{N(X)}(T)) \cap X^{-1}(r) = A \cap X^{-1}(r) \neq \emptyset$ by assumption, and

$$\begin{aligned} \text{gamb}(T) &= \bigcup_{X \in \mathcal{X}} \text{gamb} \left(\bigodot_{\substack{r \in \mathcal{R} \\ X^{-1}(r) \neq \emptyset}} X^{-1}(r)r \right) \\ &= \bigcup_{X \in \mathcal{X}} \left\{ \bigoplus_{\substack{r \in \mathcal{R} \\ X^{-1}(r) \neq \emptyset}} X^{-1}(r)r \right\} = \bigcup_{X \in \mathcal{X}} \{X\} = \mathcal{X} \end{aligned}$$

which establishes (A). \square

A.3. Proof of Lemma 15.

Proof. In this proof, A is a non-empty event and all gambles are A -consistent.

Property 2 \implies Property 5. Let $\mathcal{X}_1, \dots, \mathcal{X}_n$ be non-empty finite sets of gambles, and let $\mathcal{X} = \bigcup_{i=1}^n \mathcal{X}_i$. If $\text{opt}(\mathcal{X}|A) \cap \mathcal{X}_k \neq \emptyset$, then $\text{opt}(\mathcal{X}_k|A) = \text{opt}(\mathcal{X}|A) \cap \mathcal{X}_k$. Hence,

$$\text{opt}(\mathcal{X}|A) = \bigcup_{k=1}^n \text{opt}(\mathcal{X}|A) \cap \mathcal{X}_k = \bigcup_{\substack{k=1 \\ \mathcal{X}_k \cap \text{opt}(\mathcal{X}|A) \neq \emptyset}}^n \text{opt}(\mathcal{X}_k|A)$$

Property 5 \implies Property 4. Immediate.

Property 4 \implies Property 6. Define $X \succeq_A Y$ if $X \in \text{opt}(\{X, Y\}|A)$. First, we prove that \succeq_A is a total preorder (i.e. total, reflexive, and transitive). Clearly, \succeq_A is total since $X \in \text{opt}(\{X, Y\}|A)$ or $Y \in \text{opt}(\{X, Y\}|A)$, hence $X \succeq_A Y$ or $Y \succeq_A X$, for all normal form decisions X and Y . Obviously, \succeq_A is reflexive. Is \succeq_A transitive? Suppose $X \succeq_A Y$ and $Y \succeq_A Z$.

By Property 4,

$$\text{opt}(\{X, Y, Z\}|A) = \begin{cases} \text{opt}(\{Y, Z\}|A), & \text{or} \\ \{X\}, & \text{or} \\ \text{opt}(\{Y, Z\}|A) \cup \{X\}. \end{cases}$$

Since, $Y \succeq_A Z$, it follows that $\{X, Y\} \cap \text{opt}(\{X, Y, Z\}|A) \neq \emptyset$.

Again, by Property 4,

$$\text{opt}(\{X, Y, Z\}|A) = \begin{cases} \text{opt}(\{X, Y\}|A), & \text{or} \\ \{Z\}, & \text{or} \\ \text{opt}(\{X, Y\}|A) \cup \{Z\}. \end{cases}$$

The case $\text{opt}(\{X, Y, Z\}|A) = \{Z\}$ cannot occur however, because we just showed that $\{X, Y\} \cap \text{opt}(\{X, Y, Z\}|A) \neq \emptyset$. Hence, because $X \in \text{opt}(\{X, Y\}|A)$, it follows that $X \in \text{opt}(\{X, Y, Z\}|A)$.

Once more by Property 4,

$$\text{opt}(\{X, Y, Z\}|A) = \begin{cases} \text{opt}(\{X, Z\}|A), & \text{or} \\ \{Y\}, & \text{or} \\ \text{opt}(\{X, Z\}|A) \cup \{Y\}. \end{cases}$$

We just showed that $X \in \text{opt}(\{X, Y, Z\}|A)$, hence the second case cannot occur, and it can only be that also $X \in \text{opt}(\{X, Z\}|A)$, establishing $X \succeq_A Z$.

Finally, we prove that

$$\text{opt}(\mathcal{X}|A) = \{X \in \mathcal{X} : (\forall Y \in \mathcal{X})(X \succeq_A Y)\},$$

or equivalently, we prove for any $X \in \mathcal{X}$ that $X \in \text{opt}(\mathcal{X}|A)$ if and only if $X \in \text{opt}(\{X, Y\}|A)$ for all $Y \in \mathcal{X}$.

Indeed, by Property 4, for any X and Y in \mathcal{X} , it holds that

$$\text{opt}(\mathcal{X}|A) = \begin{cases} \text{opt}(\{X, Y\}|A), & \text{or} \\ \text{opt}(\mathcal{X} \setminus \{X, Y\}|A), & \text{or} \\ \text{opt}(\{X, Y\}|A) \cup \text{opt}(\mathcal{X} \setminus \{X, Y\}|A). \end{cases}$$

and hence, if $X \in \text{opt}(\mathcal{X}|A)$ then the second option is impossible and therefore $X \in \text{opt}(\{X, Y\}|A)$ for all $Y \in \mathcal{X}$.

Conversely, again by Property 4

$$\text{opt}(\mathcal{X}|A) = \text{opt}\left(\bigcup_{Y \in \mathcal{X}} \{X, Y\} \middle| A\right) = \bigcup_{Y \in \mathcal{Y}} \text{opt}(\{X, Y\}|A)$$

for some subset \mathcal{Y} of \mathcal{X} , and hence, if $X \in \text{opt}(\{X, Y\}|A)$ for all $Y \in \mathcal{X}$, then $X \in \text{opt}(\mathcal{X}|A)$.

Property 6 \implies Property 2. Assume that $\text{opt}(\mathcal{X}|A) \cap \mathcal{Y} \neq \emptyset$. This means that there must be an $Y^* \in \mathcal{Y}$ such that $Y^* \succeq_A X$ for all $X \in \mathcal{X}$. Clearly, $Y^* \in \text{opt}(\mathcal{Y}|A)$. But, for all $Y \in \text{opt}(\mathcal{Y}|A)$ it must also hold that $Y \succeq_A Y^*$, and hence $Y \succeq_A X$ for all $X \in \mathcal{X}$ as \succeq_A is transitive. We conclude:

$$\text{opt}(\mathcal{Y}|A) = \{Y \in \mathcal{Y} : (\forall X \in \mathcal{X})(Y \succeq_A X)\} = \text{opt}(\mathcal{X}|A) \cap \mathcal{Y}$$

□

A.4. Proof of Lemma 16.

Proof. If N is the root of T , then the statement is trivial. If $N \in \text{ch}(T)$, then the statement follows from (i). Otherwise, N must belong to $\text{st}_K(T)$ for some $K \in \text{ch}(T)$.

First, note that K is a node of at least one element of $\text{norm}(T)$. Indeed, it is given that N is a node of at least one element, say U , of $\text{norm}(T)$. Then, obviously, K must also be a node of U , simply because any node on the unique path within T between the root of T and N must be a node of U , and one of those nodes is K . So, K is a node of an element of $\text{norm}(T)$ (namely, U).

Secondly, note that N is also a node of at least one element of $\text{norm}(\text{st}_K(T))$. Indeed, N is a node of an element of $\text{norm}(T)$, and hence, in particular also of $\text{st}_K(\text{norm}(T))$. But, by (i), and the fact that K is a node of at least one element of $\text{norm}(T)$ (as just proven), it follows that $\text{st}_K(\text{norm}(T)) = \text{norm}(\text{st}_K(T))$. Hence, N is also a node of at least one element of $\text{norm}(\text{st}_K(T))$.

Combining everything, it follows that

$$\text{norm}(\text{st}_N(T)) = \text{norm}(\text{st}_N(\text{st}_K(T)))$$

so, by (ii), and because N is in at least one element of $\text{norm}(\text{st}_K(T))$,

$$= \text{st}_N(\text{norm}(\text{st}_K(T)))$$

hence, by (i), and since K is in at least one element of $\text{norm}(T)$,

$$= \text{st}_N(\text{st}_K(\text{norm}(T))) = \text{st}_N(\text{norm}(T)).$$

□

A.5. Proof of Lemma 17.

Proof. The statement is trivial if $n = 1$ (because, in that case, $A_1 = \Omega$). Let us prove the statement also in case $n \geq 2$.

Let $\mathcal{X} = \bigoplus_{i=1}^n A_i \mathcal{X}_i$. Consider any $k \in \{1, \dots, n\}$ and let

$$\mathcal{Z}_k = \bigoplus_{j \neq k} A'_j \mathcal{X}_j$$

where $(A'_j)_{j \neq k}$ forms an arbitrary partition of Ω such that $\overline{A}_k \cap A'_j = A_j$ for all $j \neq k$. Clearly, \mathcal{Z}_k is $\overline{A}_k \cap B$ -consistent because we can trivially find a consistent decision tree T with $\text{ev}(T) = \overline{A}_k \cap B$ and $\text{gamb}(T) = \mathcal{Z}_k$, using the $A_j \cap B$ -consistency (and hence, $\overline{A}_k \cap A'_j \cap B$ -consistency) of each \mathcal{X}_j for $j \neq k$.

Now, observe that by construction of \mathcal{Z}_k ,

$$\mathcal{X} = A_k \mathcal{X}_k \oplus \overline{A}_k \mathcal{Z}_k = \bigcup_{Z_k \in \mathcal{Z}_k} (A_k \mathcal{X}_k \oplus \overline{A}_k Z_k).$$

Note that \mathcal{X} is B -consistent (indeed, because each \mathcal{X}_i is $A_i \cap B$ -consistent, we can trivially find a consistent decision tree T with $\text{ev}(T) = B$ and $\text{gamb}(T) = \mathcal{X}$).

Since Property 2 holds, Property 4 holds as well by Lemma 15. So, if we apply $\text{opt}(\cdot|B)$ on both sides of the above equality, then it follows from Property 4 that

$$\text{opt}(\mathcal{X}|B) = \bigcup_{Z_k \in \mathcal{Z}_k^*} \text{opt}(A_k \mathcal{X}_k \oplus \overline{A}_k Z_k|B),$$

for some $\mathcal{Z}_k^* \subseteq \mathcal{Z}_k$. By Property 3,

$$\begin{aligned} &= \bigcup_{Z_k \in \mathcal{Z}_k^*} (A_k \text{opt}(\mathcal{X}_k|A_k \cap B) \oplus \overline{A}_k Z_k) \\ (18) \quad &= A_k \text{opt}(\mathcal{X}_k|A_k \cap B) \oplus \overline{A}_k \mathcal{Z}_k^* \end{aligned}$$

Since this holds for each $k \in \{1, \dots, n\}$, we arrive at Eq. (9), by Lemma 43. □

We used the following lemma.

Lemma 43. *Let A_1, \dots, A_n be a finite partition of Ω , $n \geq 2$. Let \mathcal{X} , and $\mathcal{X}_1, \mathcal{Z}_1, \dots, \mathcal{X}_n, \mathcal{Z}_n$ be non-empty finite sets of gambles. If*

$$\mathcal{X} = A_k \mathcal{X}_k \oplus \overline{A}_k \mathcal{Z}_k$$

for all $k \in \{1, \dots, n\}$, then

$$\mathcal{X} = \bigoplus_{i=1}^n A_i \mathcal{X}_i.$$

Proof. Let us prove the implication by induction.

The implication holds for $n = 2$. Indeed, suppose that

$$\mathcal{X} = A_1\mathcal{X}_1 \oplus \overline{A}_1\mathcal{Z}_1 = A_2\mathcal{X}_2 \oplus \overline{A}_2\mathcal{Z}_2.$$

By multiplying all elements of these sets with A_2 , and noting that $\overline{A}_1 = A_2$ and $\overline{A}_2 = A_1$, we find that

$$\overline{A}_1\mathcal{Z}_1 = A_2\mathcal{X}_2$$

which establishes the base step.

For the induction step, assume that the implication holds for $n = m$. We prove that the implication also holds for $n = m + 1$. Suppose that

$$\begin{aligned} \mathcal{X} &= A_1\mathcal{X}_1 \oplus \overline{A}_1\mathcal{Z}_1 \\ \mathcal{X} &= A_2\mathcal{X}_2 \oplus \overline{A}_2\mathcal{Z}_2 \\ &\vdots \\ \mathcal{X} &= A_{m+1}\mathcal{X}_{m+1} \oplus \overline{A}_{m+1}\mathcal{Z}_{m+1} \end{aligned}$$

First, note that the equality $A_1\mathcal{X}_1 \oplus \overline{A}_1\mathcal{Z}_1 = A_2\mathcal{X}_2 \oplus \overline{A}_2\mathcal{Z}_2$ implies in particular that (by multiplying all elements of these sets with \overline{A}_1)

$$\overline{A}_1\mathcal{Z}_1 = A_2\mathcal{X}_2 \oplus (\overline{A}_1 \cap \overline{A}_2)\mathcal{Z}_2$$

Hence,

$$\begin{aligned} \mathcal{X} &= A_1\mathcal{X}_1 \oplus \overline{A}_1\mathcal{Z}_1 = A_1\mathcal{X}_1 \oplus \overline{A}_1(A_2\mathcal{X}_2 \oplus (\overline{A}_1 \cap \overline{A}_2)\mathcal{Z}_2) \\ &= A_1\mathcal{X}_1 \oplus A_2\mathcal{X}_2 \oplus (\overline{A}_1 \cap \overline{A}_2)\mathcal{Z}_2 \\ &= (A_1 \cup A_2)(A_1\mathcal{X}_1 \oplus A_2\mathcal{X}_2) \oplus (\overline{A}_1 \cap \overline{A}_2)\mathcal{Z}_2 \end{aligned}$$

With $A_1^* = A_1 \cup A_2$, $\mathcal{X}_1^* = A_1\mathcal{X}_1 \oplus A_2\mathcal{X}_2$, and $\mathcal{Z}_1^* = \mathcal{Z}_2$, we now have that

$$\begin{aligned} \mathcal{X} &= A_1^*\mathcal{X}_1^* \oplus \overline{A}_1^*\mathcal{Z}_1^* \\ \mathcal{X} &= A_3\mathcal{X}_3 \oplus \overline{A}_3\mathcal{Z}_3 \\ &\vdots \\ \mathcal{X} &= A_{m+1}\mathcal{X}_{m+1} \oplus \overline{A}_{m+1}\mathcal{Z}_{m+1} \end{aligned}$$

By assumption, the implication to be proven already holds for $n = m$, so, from the above equalities, it follows that

$$\mathcal{X} = A_1^*\mathcal{X}_1^* \oplus A_3\mathcal{X}_3 \oplus \dots \oplus A_{m+1}\mathcal{X}_{m+1}$$

and, by construction of A_1^* and \mathcal{X}_1^* ,

$$= \bigoplus_{i=1}^{m+1} A_i\mathcal{X}_i$$

which proves the induction step. \square

A.6. Proof of Lemma 18.

Proof. By Eq. (5), Eq. (10) holds if and only if

$$\text{gamb}(\text{nfd}(T_i)) \cap \text{opt}(\text{gamb}(T)|\text{ev}(T)) \neq \emptyset,$$

or equivalently, if and only if there is a normal form decision $U \in \text{nfd}(T_i)$ such that $\text{gamb}(U) \subseteq \text{opt}(\text{gamb}(T)|\text{ev}(T))$ (remember that $\text{gamb}(U)$ is a singleton).

But, by definition of the gamb operator, it also holds that $\text{gamb}(U) = \text{gamb}(\sqcup U)$. Hence, Eq. (10) holds if and only if there is a normal form decision $U \in \text{nfd}(T_i)$ such that $\text{gamb}(\sqcup U) \subseteq \text{opt}(\text{gamb}(T)|\text{ev}(T))$, or equivalently, if and only if there is a normal form decision $V \in \text{nfd}(T)$ which contains the node N_i , such that $\text{gamb}(V) \subseteq \text{opt}(\text{gamb}(T)|\text{ev}(T))$.

By definition of norm_{opt} , this is equivalent to stating that Eq. (10) holds if and only if N_i is in at least one element of $\text{norm}_{\text{opt}}(T)$. \square

A.7. Proof of Lemma 19.

Proof. Assume that Eq. (11) holds.

First, consider a normal form decision $U \in \bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i)$. Obviously,

$$\text{gamb}(U) \subseteq \text{gamb}\left(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i)\right)$$

so, by the definition of gamb, Eq. (7) in particular,

$$= \bigoplus_{i=1}^n E_i \text{gamb}(\text{norm}_{\text{opt}}(T_i))$$

and hence, by Eq. (11),

$$= \text{gamb}(\text{norm}_{\text{opt}}(T)).$$

So, there exists a normal form decision $V \in \text{norm}_{\text{opt}}(T)$ such that $\text{gamb}(V) = \text{gamb}(U)$. Since $U \in \text{nfd}(T)$, by definition of norm_{opt} we have $U \in \text{norm}_{\text{opt}}(T)$. So we have shown that

$$\text{norm}_{\text{opt}}(T) \supseteq \bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i).$$

Next, consider a normal form decision $U \in \text{norm}_{\text{opt}}(T)$. We know by Eq. (11) that

$$\text{gamb}(U) \subseteq \bigoplus_{i=1}^n E_i \text{gamb}(\text{norm}_{\text{opt}}(T_i)).$$

We can write $U = \bigodot_{i=1}^n E_i U_i$, where $U_i \in \text{nfd}(T_i)$, so

$$\bigoplus_{i=1}^n E_i \text{gamb}(U_i) \subseteq \bigoplus_{i=1}^n E_i \text{gamb}(\text{norm}_{\text{opt}}(T_i)).$$

Consider any k and any normal form decision $V_k \in \text{norm}_{\text{opt}}(T_k)$. The above equation, and Eq. (6), tell us that we can choose each V_k such that $E_k \text{gamb}(V_k) = E_k \text{gamb}(U_k)$. Of course, because $V_k \in \text{norm}_{\text{opt}}(T_k)$,

$$\text{gamb}(V_k) \subseteq \text{opt}(\text{gamb}(T_k)|\text{ev}(T_k)).$$

We wish to establish that also $\text{gamb}(U_k) \subseteq \text{opt}(\text{gamb}(T_k)|\text{ev}(T_k))$.

Indeed, this follows from Property 1: observe that both singletons $\text{gamb}(U_k)$ and $\text{gamb}(V_k)$ are subsets of $\text{gamb}(T_k)$, $E_k \text{gamb}(U_k) = E_k \text{gamb}(V_k)$, and $\text{gamb}(V_k) \subseteq \text{opt}(\text{gamb}(T_k)|\text{ev}(T_k))$. Consistency of T confirms that $\text{gamb}(T_k)$ is $\text{ev}(T_k)$ -consistent. Hence, Property 1 applies, and

$$\text{gamb}(U_k) \subseteq \text{opt}(\text{gamb}(T_k)|\text{ev}(T_k)).$$

Therefore, $U_k \in \text{norm}_{\text{opt}}(T_k)$ by definition of $\text{norm}_{\text{opt}}(T_i)$. Since this holds for any k , we conclude that $U \in \bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i)$. So, we have shown that also

$$\text{norm}_{\text{opt}}(T) \subseteq \bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i).$$

□

A.8. Proof of Lemma 20.

Proof. Assume that Eq. (12) holds.

Consider any normal form decision $V \in \text{nfd}(\bigsqcup_{i \in \mathcal{I}} \text{norm}_{\text{opt}}(T_i))$. By definition of gamb ,

$$\text{gamb}(V) \subseteq \bigcup_{i \in \mathcal{I}} \text{gamb}(\text{norm}_{\text{opt}}(T_i))$$

and, by Eq. (12),

$$= \text{gamb}(\text{norm}_{\text{opt}}(T)).$$

Hence, by definition of norm_{opt} , and the obvious fact that $V \in \text{nfd}(T)$, it follows that $V \in \text{norm}_{\text{opt}}(T)$. So we have shown that

$$\text{norm}_{\text{opt}}(T) \supseteq \text{nfd}\left(\bigsqcup_{i \in \mathcal{I}} \text{norm}_{\text{opt}}(T_i)\right).$$

Conversely, let $V \in \text{norm}_{\text{opt}}(T)$. Then, again by Eq. (12),

$$\text{gamb}(V) \subseteq \text{gamb}(\text{norm}_{\text{opt}}(T)) = \bigcup_{i \in \mathcal{I}} \text{gamb}(\text{norm}_{\text{opt}}(T_i)).$$

Now $V = \sqcup U$ where $U \in \text{nfd}(T_i)$ for some $i \in \mathcal{I}$. We want to show that $U \in \text{norm}_{\text{opt}}(T_i)$.

Indeed, let X be the gamble corresponding to V , and also U ,

$$\text{gamb}(V) = \text{gamb}(U) = \{X\}.$$

Because $V \in \text{norm}_{\text{opt}}(T)$, we know that $X \in \text{opt}(\text{gamb}(T)|\text{ev}(T))$. It is established that $U \in \text{norm}_{\text{opt}}(T_i)$ if we can show that $X \in \text{opt}(\text{gamb}(T_i)|\text{ev}(T_i))$. But this follows at once from Property 2, because $i \in \mathcal{I}$ (and the definition of \mathcal{I}), $\text{gamb}(T_i) \subseteq \text{gamb}(T)$, $\text{ev}(T) = \text{ev}(T_i)$, and all sets of gambles are consistent with respect to the relevant events:

$$\text{opt}(\text{gamb}(T_i)|\text{ev}(T_i)) = \text{opt}(\text{gamb}(T)|\text{ev}(T)) \cap \text{gamb}(T_i).$$

Concluding, also

$$\text{norm}_{\text{opt}}(T) \subseteq \text{nfd}\left(\bigsqcup_{i \in \mathcal{I}} \text{norm}_{\text{opt}}(T_i)\right).$$

□

A.9. Proof of Lemma 21.

Proof. Assume that norm_{opt} is subtree perfect.

We first establish Property 1. Let A be a non-empty event, and let \mathcal{X} be a non-empty finite set of A -consistent gambles such that $\{X, Y\} \subseteq \mathcal{X}$ with $AX = AY$ and $X \in \text{opt}(\mathcal{X}|A)$. We show that $Y \in \text{opt}(\mathcal{X}|A)$. If $A = \Omega$ the result is trivial, so assume $A \subset \Omega$.

Consider a consistent decision tree $T = AT_1 \odot \overline{A}T_2$, where $\text{ev}(T) = \Omega$, $\text{gamb}(T_1) = \mathcal{X}$, and T_2 is a normal form decision with $\text{gamb}(T_2) = \{Z\}$, an \overline{A} -consistent gamble (see the left tree in Fig. 6). We know by consistency of the gambles that there is such a T (see Definition 9).

Consider $U \in \text{nfd}(T_1)$ with $\text{gamb}(U) = \{X\}$ and $V \in \text{nfd}(T_1)$ with $\text{gamb}(V) = \{Y\}$. By definition of norm_{opt} , we have $U \in \text{norm}_{\text{opt}}(T_1)$. Therefore by subtree perfectness, $AU \odot \overline{A}T_2 \in \text{norm}_{\text{opt}}(T)$ and of course $AV \odot \overline{A}T_2 \in \text{norm}_{\text{opt}}(T)$. Again by subtree perfectness, $V \in \text{norm}_{\text{opt}}(T_1)$, whence $Y \in \text{opt}(\mathcal{X}|A)$.

Next, we establish Property 2. Let A be a non-empty event, and let $\mathcal{Y} \subseteq \mathcal{X}$ be non-empty finite A -consistent sets of gambles such that $\text{opt}(\mathcal{X}|A) \cap \mathcal{Y} \neq \emptyset$. We show that $\text{opt}(\mathcal{Y}|A) = \text{opt}(\mathcal{X}|A) \cap \mathcal{Y}$.

Let $T = T_1 \sqcup T_2$ be a consistent decision tree with $\text{ev}(T) = A$, $\text{gamb}(T_1) = \mathcal{Y}$ and $\text{gamb}(T_2) = \mathcal{X}$ (see the right tree in Fig. 6). We know by consistency of the gambles that there is such a T .

Let N be the decision node at the root of T_1 . By subtree perfectness, we have

$$\text{gamb}(\text{st}_N(\text{norm}_{\text{opt}}(T))) = \text{gamb}(\text{norm}_{\text{opt}}(\text{st}_N(T))).$$

The right-hand side is equal to $\text{opt}(\mathcal{Y}|A)$. Also,

$$\begin{aligned} \text{gamb}(\text{st}_N(\text{norm}_{\text{opt}}(T))) &= \text{gamb}(\text{norm}_{\text{opt}}(T)) \cap \text{gamb}(\text{st}_N(T)) \\ &= \text{opt}(\mathcal{X}|A) \cap \mathcal{Y} \end{aligned}$$

as required.

Finally, we establish Property 3. Let A and B be events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, let \mathcal{X} be a non-empty finite $A \cap B$ -consistent set of gambles, and let Z be a $\overline{A} \cap B$ -consistent gamble.

Let $T = AT_1 \odot \overline{A}T_2$ be a consistent decision tree such that $\text{ev}(T) = B$, $\text{gamb}(T_1) = \mathcal{X}$, and $\text{gamb}(T_2) = \{Z\}$ (see the left tree in Fig. 6).

By subtree perfectness, we have (letting N be the root node of T_1)

$$\text{gamb}(\text{st}_N(\text{norm}_{\text{opt}}(T))) = \text{gamb}(\text{norm}_{\text{opt}}(\text{st}_N(T))).$$

Here, the right-hand side is $\text{opt}(\mathcal{X}|A \cap B)$, and

$$\text{gamb}(\text{st}_N(\text{norm}_{\text{opt}}(T))) = \{X \in \mathcal{X} : AX \oplus \overline{A}Z \in \text{opt}(A\mathcal{X} \oplus \overline{A}Z|B),$$

whence Property 3 follows. \square

A.10. Proof of Theorem 25. Before we prove Theorem 25, we prove a few lemmas, provide two counterexamples, and introduce a new property called path independence.

Lemma 44 (Sen [24, Proposition 17]). *A choice function opt satisfies Property 9 if and only if, for any non-empty event A and any finite family of non-empty finite*

A -consistent sets of gambles $\mathcal{X}_1, \dots, \mathcal{X}_n$,

$$\text{opt} \left(\bigcup_{i=1}^n \mathcal{X}_i \middle| A \right) \subseteq \text{opt} \left(\bigcup_{i=1}^n \text{opt}(\mathcal{X}_i | A) \middle| A \right) \subseteq \bigcup_{i=1}^n \text{opt}(\mathcal{X}_i | A).$$

Property 11. A choice function opt is said to be path independent if, for any non-empty event A , and for any finite family of non-empty finite A -consistent sets of gambles $\mathcal{X}_1, \dots, \mathcal{X}_n$,

$$\text{opt} \left(\bigcup_{i=1}^n \mathcal{X}_i \middle| A \right) = \text{opt} \left(\bigcup_{i=1}^n \text{opt}(\mathcal{X}_i | A) \middle| A \right).$$

Path independence appears frequently in the social choice literature. Plott [18] gives a detailed investigation of path independence and its possible justifications. Path independence is also equivalent to Axiom 7' of Luce and Raiffa [15, p. 289].

Lemma 45 (Sen [24, Proposition 19]). A choice function opt satisfies Properties 8 and 9 if and only if opt satisfies Property 11.

Properties 8, 9, and 11 are expressed slightly differently here than in Sen [24], who does not use the concepts of conditioning and consistency. Despite this, the proofs of Lemmas 44 and 45 proceed identically to the corresponding propositions by Sen. Also, Sen defines path independence only for pairs of subsets, but Plott [18, Theorem 1, p. 1082] shows that this type of path independence is indeed equivalent to Property 11.

Lemma 46. Let A_1, \dots, A_n be a finite partition of Ω , $n \geq 2$. Let \mathcal{X} , and $\mathcal{X}_1, \mathcal{Z}_1, \dots, \mathcal{X}_n, \mathcal{Z}_n$ be finite sets of gambles. If

$$\mathcal{X} \subseteq A_k \mathcal{X}_k \oplus \overline{A}_k \mathcal{Z}_k$$

for all $k \in \{1, \dots, n\}$, then

$$\mathcal{X} \subseteq \bigoplus_{k=1}^n A_k \mathcal{X}_k.$$

Proof. Let $X \in \mathcal{X}$, then $X \in A_k \mathcal{X}_k \oplus \overline{A}_k \mathcal{Z}_k$ for all k . Therefore, for each k , there is an $X_k \in \mathcal{X}_k$ such that $A_k X = A_k X_k$. Whence,

$$X = \bigoplus_{k=1}^n A_k X_k \in \bigoplus_{k=1}^n A_k \mathcal{X}_k.$$

□

Lemma 47. Let A_1, \dots, A_n be a finite partition of Ω . Let B be any event such that $A_i \cap B \neq \emptyset$ for all A_i . Let $\mathcal{X}_1, \dots, \mathcal{X}_n$ be a finite family of non-empty finite sets of gambles where each \mathcal{X}_i is $A_i \cap B$ -consistent. If a choice function opt satisfies Properties 9 and 10, then

$$(19) \quad \text{opt} \left(\bigoplus_{i=1}^n A_i \mathcal{X}_i \middle| B \right) \subseteq \bigoplus_{i=1}^n A_i \text{opt}(\mathcal{X}_i | A_i \cap B).$$

Proof. Let $\mathcal{X} = \bigoplus_{i=1}^n A_i \mathcal{X}_i$. Consider any $k \in \{1, \dots, n\}$ and let

$$\mathcal{Z}_k = \bigoplus_{j \neq k} A_j \mathcal{X}_j$$

where $(A'_j)_{j \neq k}$ forms an arbitrary partition of Ω such that $\overline{A}_k \cap A'_j = A_j$ for all $j \neq k$. Clearly, \mathcal{Z}_k is $\overline{A}_k \cap B$ -consistent because we can trivially find a consistent decision tree T with $\text{ev}(T) = \overline{A}_k \cap B$ and $\text{gamb}(T) = \mathcal{Z}_k$, using the $A_j \cap B$ -consistency (and hence, $\overline{A}_k \cap A'_j \cap B$ -consistency) of each \mathcal{X}_j for $j \neq k$.

Now, observe that by construction of \mathcal{Z}_k ,

$$\mathcal{X} = A_k \mathcal{X}_k \oplus \overline{A}_k \mathcal{Z}_k = \bigcup_{Z_k \in \mathcal{Z}_k} (A_k \mathcal{X}_k \oplus \overline{A}_k Z_k).$$

Note that \mathcal{X} is B -consistent (indeed, because each \mathcal{X}_i is $A_i \cap B$ -consistent, we can trivially find a consistent decision tree T with $\text{ev}(T) = B$ and $\text{gamb}(T) = \mathcal{X}$).

If we apply $\text{opt}(\cdot|B)$ on both sides of the above equality, then it follows from Lemma 44 that

$$\text{opt}(\mathcal{X}|B) \subseteq \bigcup_{Z_k \in \mathcal{Z}_k} \text{opt}(A_k \mathcal{X}_k \oplus \overline{A}_k Z_k|B)$$

and by Property 10 (once noted that \mathcal{X}_k is $A_k \cap B$ -consistent by assumption, and Z_k is $\overline{A}_k \cap B$ -consistent by construction),

$$\begin{aligned} &\subseteq \bigcup_{Z_k \in \mathcal{Z}_k} (A_k \text{opt}(\mathcal{X}_k|A_k \cap B) \oplus \overline{A}_k Z_k) \\ &= A_k \text{opt}(\mathcal{X}_k|A_k \cap B) \oplus \overline{A}_k \mathcal{Z}_k \end{aligned}$$

whence by Lemma 46,

$$\text{opt}(\mathcal{X}|B) \subseteq \bigoplus_{i=1}^n A_i \text{opt}(\mathcal{X}_i|A_i \cap B).$$

□

Lemma 48. *Let A_1, \dots, A_n be a finite partition of Ω . Let B be any event such that $A_i \cap B \neq \emptyset$ for all A_i . Let $\mathcal{X}_1, \dots, \mathcal{X}_n$ be a finite family of non-empty finite sets of gambles, where each \mathcal{X}_i is $A_i \cap B$ -consistent. If a choice function opt satisfies Properties 8, 9, and 10, then*

$$(20) \quad \text{opt} \left(\bigoplus_{i=1}^n A_i \mathcal{X}_i \middle| B \right) = \text{opt} \left(\bigoplus_{i=1}^n A_i \text{opt}(\mathcal{X}_i|A_i \cap B) \middle| B \right).$$

Proof. By Lemma 47 and the definition of opt ,

$$\text{opt} \left(\bigoplus_{i=1}^n A_i \mathcal{X}_i \middle| B \right) \subseteq \bigoplus_{i=1}^n A_i \text{opt}(\mathcal{X}_i|A_i \cap B) \subseteq \bigoplus_{i=1}^n A_i \mathcal{X}_i,$$

whence Eq. (20) follows by Property 8. □

Lemma 49. *For any consistent decision tree $T = \bigodot_{i=1}^n E_i T_i$ and any choice function opt satisfying Property 7,*

$$(21) \quad \text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{gamb} \left(\text{norm}_{\text{opt}} \left(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i) \right) \right)$$

implies

$$\text{norm}_{\text{opt}}(T) = \text{norm}_{\text{opt}} \left(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i) \right).$$

Proof. In this proof we require Eq. (6), and the following consequence of Eq. (7):

$$(22) \quad \text{gamb} \left(\text{norm}_{\text{opt}} \left(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i) \right) \right) \\ = \text{opt} \left(\bigoplus_{i=1}^n E_i \text{opt}(\text{gamb}(T_i) | \text{ev}(T) \cap E_i) \Big| \text{ev}(T) \right).$$

We first show that

$$\text{norm}_{\text{opt}}(T) \supseteq \text{norm}_{\text{opt}} \left(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i) \right).$$

Consider a normal form decision $U \in \text{norm}_{\text{opt}}(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i))$. We have, by Eq. (21),

$$\text{gamb}(U) \subseteq \text{gamb} \left(\text{norm}_{\text{opt}} \left(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i) \right) \right) = \text{gamb}(\text{norm}_{\text{opt}}(T)).$$

So, there exists a normal form decision $V \in \text{norm}_{\text{opt}}(T)$ such that $\text{gamb}(V) = \text{gamb}(U)$. Since $U \in \text{nfd}(T)$, by definition of norm_{opt} , $U \in \text{norm}_{\text{opt}}(T)$, which establishes the claim.

Next, we show that

$$\text{norm}_{\text{opt}}(T) \subseteq \text{norm}_{\text{opt}} \left(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i) \right).$$

Consider a normal form decision $U \in \text{norm}_{\text{opt}}(T)$. We know by Eq. (21) and Eq. (22) that

$$\text{gamb}(U) \subseteq \text{opt} \left(\bigoplus_{i=1}^n E_i \text{opt}(\text{gamb}(T_i) | \text{ev}(T) \cap E_i) \Big| \text{ev}(T) \right).$$

We can write $U = \bigodot_{i=1}^n E_i U_i$, where $U_i \in \text{nfd}(T_i)$, so by Eq. (4b),

$$\bigoplus_{i=1}^n E_i \text{gamb}(U_i) \subseteq \text{opt} \left(\bigoplus_{i=1}^n E_i \text{opt}(\text{gamb}(T_i) | \text{ev}(T) \cap E_i) \Big| \text{ev}(T) \right).$$

Consider normal form decisions $V_i \in \text{norm}_{\text{opt}}(T_i)$. The above equation, and Eq. (6) tell us that, for each i , we can find V_i such that $E_i \text{gamb}(V_i) = E_i \text{gamb}(U_i)$. Of course, because $V_i \in \text{norm}_{\text{opt}}(T_i)$,

$$\text{gamb}(V_i) \subseteq \text{opt}(\text{gamb}(T_i) | E_i \cap \text{ev}(T)).$$

We further have, for $V = \bigodot_{i=1}^n E_i V_i$, $\text{gamb}(V) = \text{gamb}(U)$ and $V \in \text{nfd}(T)$, and so $V \in \text{norm}_{\text{opt}}(T)$.

If we can establish that

$$(23) \quad \text{gamb}(U_i) \subseteq \text{opt}(\text{gamb}(T_i) | E_i \cap \text{ev}(T)),$$

then, by definition of norm_{opt} and because $U_i \in \text{nfd}(T_i)$, it follows that $U_i \in \text{norm}_{\text{opt}}(T_i)$. So, in that case, there is a $V \in \text{norm}_{\text{opt}}(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i))$ such that $\text{gamb}(V) = \text{gamb}(U)$, and $U \in \text{nfd}(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i))$. Therefore by definition of norm_{opt} , we will have $U \in \text{norm}_{\text{opt}}(\bigodot_{i=1}^n E_i \text{norm}_{\text{opt}}(T_i))$, establishing the desired result.

We show that Eq. (23) indeed holds by Property 7. When $n = 1$ the result is trivial, so assume $n \geq 2$. Observe that both singletons $\text{gamb}(U_i)$ and $\text{gamb}(V_i)$ are subsets of $\text{gamb}(T_i)$, $E_i \text{gamb}(U_i) = E_i \text{gamb}(V_i)$, and $\text{gamb}(V_i) \subseteq \text{opt}(\text{gamb}(T_i)|E_i \cap \text{ev}(T))$. Further, $\text{gamb}(T_i)$ is $E_i \cap \text{ev}(T)$ -consistent. We are almost ready to apply Property 7.

We know that

$$\begin{aligned} \text{gamb}(V) &= \bigoplus_{i=1}^n E_i \text{gamb}(V_i) \subseteq \text{opt}(\text{gamb}(T)|\text{ev}(T)) \\ &= \text{opt}\left(\bigoplus_{i=1}^n E_i \text{gamb}(T_i) \middle| \text{ev}(T)\right). \end{aligned}$$

Letting

$$Z = (E_1 \cup E_2) \text{gamb}(V_2) \oplus E_3 \text{gamb}(V_3) \oplus \dots \oplus E_n \text{gamb}(V_n)$$

and

$$\mathcal{Z} = (E_1 \cup E_2) \text{gamb}(T_2) \oplus E_3 \text{gamb}(T_3) \oplus \dots \oplus E_n \text{gamb}(T_n),$$

we see that $Z \in \mathcal{Z}$ and \mathcal{Z} is $\overline{E}_1 \cap \text{ev}(T)$ -consistent. Further,

$$E_1 \text{gamb}(V_1) \oplus \overline{E}_1 Z = \bigoplus_{i=1}^n E_i \text{gamb}(V_i) = \text{gamb}(V)$$

and

$$E_1 \text{gamb}(T_1) \oplus \overline{E}_1 \mathcal{Z} = \bigoplus_{i=1}^n E_i \text{gamb}(T_i).$$

We see that

$$E_1 \text{gamb}(V) \oplus \overline{E}_1 Z \subseteq \text{opt}(E_1 \text{gamb}(T_1) \oplus \overline{E}_1 \mathcal{Z} | \text{ev}(T)).$$

Hence we have found a \mathcal{Z} and a $Z \in \mathcal{Z}$ required to apply Property 7. Finally, by $E_1 \text{gamb}(U_1) = E_1 \text{gamb}(V_1)$, and Property 7, we have

$$\text{gamb}(U_1) \subseteq \text{opt}(\text{gamb}(T_1)|E_1 \cap \text{ev}(T)).$$

This argument applies for any index i and therefore Eq. (23) has been shown, establishing the result. \square

Lemma 50. *For any consistent decision tree $T = \bigsqcup_{i=1}^n T_i$, and any choice function opt satisfying Property 9,*

$$(24) \quad \text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{gamb}\left(\text{norm}_{\text{opt}}\left(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i)\right)\right)$$

implies

$$\text{norm}_{\text{opt}}(T) = \text{norm}_{\text{opt}}\left(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i)\right).$$

Proof. In this proof we require of Eq. (6). For clarity, let $A = \text{ev}(T) = \text{ev}(T_i)$. We first show that

$$\text{norm}_{\text{opt}}(T) \supseteq \text{norm}_{\text{opt}}\left(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i)\right).$$

Consider a normal form decision $U \in \text{norm}_{\text{opt}}(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i))$. To show that $U \in \text{norm}_{\text{opt}}(T)$, we must show that $U \in \text{nfd}(T)$ and $\text{gamb}(U) \subseteq \text{gamb}(\text{norm}_{\text{opt}}(T))$. The former is obvious, and the latter is established by Eq. (24):

$$\text{gamb}(U) \subseteq \text{gamb}\left(\text{norm}_{\text{opt}}\left(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i)\right)\right) = \text{gamb}(\text{norm}_{\text{opt}}(T)).$$

Next we show that

$$\text{norm}_{\text{opt}}(T) \subseteq \text{norm}_{\text{opt}}\left(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i)\right).$$

Let $U \in \text{norm}_{\text{opt}}(T)$. To show that $U \in \text{norm}_{\text{opt}}(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i))$ we must show that $U \in \text{nfd}(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i))$ and that

$$\text{gamb}(U) \subseteq \text{gamb}\left(\text{norm}_{\text{opt}}\left(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i)\right)\right).$$

The latter requirement follows immediately from Eq. (24):

$$\text{gamb}(U) \subseteq \text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{gamb}\left(\text{norm}_{\text{opt}}\left(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i)\right)\right).$$

We now prove that $U \in \text{nfd}(\bigsqcup_{i=1}^n \text{norm}_{\text{opt}}(T_i))$. Let V be U with the root node removed, that is, $U = \sqcup V$. Clearly, for some k , $V \in \text{nfd}(T_k)$. It suffices to show that $V \in \text{norm}_{\text{opt}}(T_k)$. Let $\{X\} = \text{gamb}(U) = \text{gamb}(V)$. We know that $X \in \text{opt}(\text{gamb}(T)|A)$, and also that $X \in \text{gamb}(T_k)$. If we can prove that $X \in \text{gamb}(\text{norm}_{\text{opt}}(T_k)) = \text{opt}(\text{gamb}(T_k)|A)$, then $V \in \text{norm}_{\text{opt}}(T_k)$. Indeed, using Property 9 and $\text{gamb}(T_k) \subseteq \text{gamb}(T)$ we have

$$\text{opt}(\text{gamb}(T_k)|A) \supseteq \text{gamb}(T_k) \cap \text{opt}(\text{gamb}(T)|A).$$

So we have shown that indeed $X \in \text{opt}(\text{gamb}(T_k)|A)$, establishing the claim. \square

Lemma 51. *If $\text{back}_{\text{opt}}(T) = \text{norm}_{\text{opt}}(T)$ for any consistent decision tree T , then opt satisfies Property 7.*

Proof. Let A and B be non-empty events, and \mathcal{X}, \mathcal{Z} be non-empty finite sets of gambles, such that the following properties hold: $A \cap B \neq \emptyset$, $\overline{A} \cap B \neq \emptyset$, \mathcal{X} is $A \cap B$ -consistent, \mathcal{Z} is $\overline{A} \cap B$ -consistent, and there are $X, Y \in \mathcal{X}$ such that $AX = AY$, $X \in \text{opt}(\mathcal{X}|A \cap B)$, and $AX \oplus \overline{A}Z \in \text{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B)$ for at least one $Z \in \mathcal{Z}$. If it is not possible to construct such a situation, then opt satisfies Property 7 automatically. Otherwise, to prove that Property 7 holds, we must show that $Y \in \text{opt}(\mathcal{X}|A \cap B)$.

Consider a consistent decision tree $T = AT_1 \odot \overline{A}T_2$, where $\text{ev}(T) = B$, $\text{gamb}(T_1) = \mathcal{X}$, and $\text{gamb}(T_2) = \mathcal{Z}$. Since \mathcal{X} is $A \cap B$ -consistent and \mathcal{Z} is $\overline{A} \cap B$ -consistent, we know from Definition 9 that there is such a T . We have $\text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{opt}(\text{gamb}(T)|B) = \text{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B)$. So, $AX \oplus \overline{A}Z \in \text{gamb}(\text{norm}_{\text{opt}}(T))$, and of course $AX \oplus \overline{A}Z = AY \oplus \overline{A}Z$.

Therefore, any normal form decision in $\text{nfd}(T)$ that induces the gamble $AY \oplus \overline{A}Z$ must be in $\text{norm}_{\text{opt}}(T)$. In particular, by Lemma 42 there is a normal form decision $U \in \text{nfd}(T_1)$ such that $\text{gamb}(U) = \{Y\}$, and a normal form decision $V \in \text{nfd}(T_2)$ such that $\text{gamb}(V) = \{Z\}$. So $AU \odot \overline{A}V \in \text{nfd}(T)$ and $\text{gamb}(AU \odot \overline{A}V) = \{AY \oplus \overline{A}Z\}$. Indeed, because $AX \oplus \overline{A}Z \in \text{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B)$, it follows that $AU \odot \overline{A}V \in$

$\text{norm}_{\text{opt}}(T) = \text{back}_{\text{opt}}(T)$. By definition, $\text{back}_{\text{opt}}(T) = \text{norm}_{\text{opt}}(A \text{back}_{\text{opt}}(T_1) \odot \overline{A} \text{back}_{\text{opt}}(T_2))$, and so it must hold that $U \in \text{back}_{\text{opt}}(T_1) = \text{norm}_{\text{opt}}(T_1)$. Whence, $\text{gamb}(U) \subseteq \text{gamb}(\text{norm}_{\text{opt}}(T_1)) = \text{opt}(\mathcal{X}|A \cap B)$. Since $\text{gamb}(U) = \{Y\}$, we have $Y \in \text{opt}(\mathcal{X}|A \cap B)$, establishing Property 7. \square

Lemma 52. *If $\text{gamb}(\text{back}_{\text{opt}}(T)) = \text{gamb}(\text{norm}_{\text{opt}}(T))$ for any consistent decision tree T , then opt satisfies Property 11.*

Proof. Let A be a non-empty event, and $\mathcal{X}_1, \dots, \mathcal{X}_n$ be non-empty finite sets of A -consistent gambles. Let $T = \bigsqcup_{i=1}^n T_i$ be a consistent decision tree, with $\text{gamb}(T_i) = \mathcal{X}_i$ for each i , and where $\text{ev}(T) = A$. The existence of T is assured by A -consistency of \mathcal{X} (see Definition 9). We have

$$\begin{aligned} \text{opt} \left(\bigcup_{i=1}^n \mathcal{X}_i \middle| A \right) &= \text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{gamb}(\text{back}_{\text{opt}}(T)) \\ &= \text{gamb} \left(\text{norm}_{\text{opt}} \left(\bigsqcup_{i=1}^n \text{back}_{\text{opt}}(T_i) \right) \right) \end{aligned}$$

From Eq. (6), we have $\text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{opt}(\text{gamb}(T)|A)$. Similarly, with repeated applications of Eq. (6) and Eq. (8),

$$\begin{aligned} &= \text{opt} \left(\text{gamb} \left(\bigsqcup_{i=1}^n \text{back}_{\text{opt}}(T_i) \right) \middle| A \right) \\ &= \text{opt} \left(\bigcup_{i=1}^n \text{gamb}(\text{back}_{\text{opt}}(T_i)) \middle| A \right) \\ &= \text{opt} \left(\bigcup_{i=1}^n \text{gamb}(\text{norm}_{\text{opt}}(T_i)) \middle| A \right) \\ &= \text{opt} \left(\bigcup_{i=1}^n \text{opt}(\text{gamb}(T_i)|A) \middle| A \right). \end{aligned}$$

Finally, we note that $\text{gamb}(T) = \bigcup_{i=1}^n \mathcal{X}_i$ and $\text{gamb}(T_i) = \mathcal{X}_i$. Therefore,

$$= \text{opt} \left(\bigcup_{i=1}^n \text{opt}(\mathcal{X}_i|A) \middle| A \right).$$

\square

Lemma 53. *If $\text{gamb}(\text{back}_{\text{opt}}(T)) = \text{gamb}(\text{norm}_{\text{opt}}(T))$ for any consistent decision tree T , then opt satisfies Property 10.*

Proof. Let A and B be non-empty events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, let \mathcal{X} be a non-empty finite set of $A \cap B$ -consistent gambles, and let Z be an $\overline{A} \cap B$ -consistent gamble. Let $T = AT_1 \odot \overline{A}T_2$ be a consistent decision tree, where $\text{ev}(T) = B$, $\text{gamb}(T_1) = \mathcal{X}$, and $\text{gamb}(T_2) = \{Z\}$. The existence of T is assured by $A \cap B$ -consistency of \mathcal{X} and $\overline{A} \cap B$ -consistency of $\{Z\}$. By assumption,

$$\begin{aligned} \text{gamb}(\text{norm}_{\text{opt}}(T)) &= \text{gamb}(\text{back}_{\text{opt}}(T)) \\ &= \text{gamb}(\text{norm}_{\text{opt}}(A \text{back}_{\text{opt}}(T_1) \odot \overline{A}T_2)). \end{aligned}$$

From Eq. (6), we have $\text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{opt}(\text{gamb}(T)|B)$. Similarly, with repeated applications of Eq. (6) and Eq. (7),

$$\begin{aligned} \text{gamb}(\text{norm}_{\text{opt}}(A \text{back}_{\text{opt}}(T_1) \odot \overline{AT_2})) &= \text{opt}(\text{gamb}(A \text{back}_{\text{opt}}(T_1) \odot \overline{AT_2})|B) \\ &= \text{opt}(A \text{gamb}(\text{back}_{\text{opt}}(T_1)) \oplus \overline{AZ}|B) \\ &= \text{opt}(A \text{gamb}(\text{norm}_{\text{opt}}(T_1)) \oplus \overline{AZ}|B) \\ &= \text{opt}(A \text{opt}(\text{gamb}(T_1)|A \cap B) \oplus \overline{AZ}|B). \end{aligned}$$

Finally we note that $\text{gamb}(T) = A\mathcal{X} \oplus \overline{AZ}$, and $\text{gamb}(T_1) = \mathcal{X}$. Therefore,

$$\text{opt}(A\mathcal{X} \oplus \overline{AZ}|B) = \text{opt}(A \text{opt}(\mathcal{X}|A \cap B) \oplus \overline{AZ}|B) \subseteq A \text{opt}(\mathcal{X}|A \cap B) \oplus \overline{AZ}.$$

□

We now prove Theorem 25.

Proof of Theorem 25. “only if”. By Lemmas 51, 52, and 53, we see that satisfying backward induction implies Properties 7, 10, and 11. Lemma 45 completes the proof.

“if”. We prove this part by structural induction on the tree. In the base step, we prove that the implication holds for consistent decision trees which consist of only a single node. In the induction step, we prove that if the implication holds for the subtrees at every child of the root node, then the implication also holds for the whole tree.

First, if the decision tree T has only a single node, and hence, a reward at the root and no further children, then by definition (Eq. (4a) in particular) we have $\text{back}_{\text{opt}}(T) = \text{norm}_{\text{opt}}(T)$.

Next, suppose T is consistent and has a chance node as its root, that is, $T = \bigodot_{i=1}^n E_i T_i$. By the induction hypothesis, we know that for every T_i ,

$$(25) \quad \text{gamb}(\text{back}_{\text{opt}}(T_i)) = \text{gamb}(\text{norm}_{\text{opt}}(T_i)).$$

We show that $\text{back}_{\text{opt}}(T) = \text{norm}_{\text{opt}}(T)$. By Lemma 49, it therefore suffices to show that $\text{gamb}(\text{norm}_{\text{opt}}(T)) = \text{gamb}(\text{back}_{\text{opt}}(T))$. By Eq. (6) and the definition of gamb,

$$\begin{aligned} \text{gamb}(\text{norm}_{\text{opt}}(T)) &= \text{opt}(\text{gamb}(T)|\text{ev}(T)) \\ &= \text{opt}\left(\text{gamb}\left(\bigodot_{i=1}^n E_i T_i\right)\middle|\text{ev}(T)\right) \\ &= \text{opt}\left(\bigoplus_{i=1}^n E_i \text{gamb}(T_i)\middle|\text{ev}(T)\right), \end{aligned}$$

and by Eq. (25), Eq. (6), and the definition of gamb ,

$$\begin{aligned}
\text{gamb}(\text{back}_{\text{opt}}(T)) &= \text{gamb} \left(\text{norm}_{\text{opt}} \left(\bigodot_{i=1}^n E_i \text{back}_{\text{opt}}(T_i) \right) \right) \\
&= \text{opt} \left(\text{gamb} \left(\bigodot_{i=1}^n E_i \text{back}_{\text{opt}}(T_i) \right) \middle| \text{ev}(T) \right) \\
&= \text{opt} \left(\bigoplus_{i=1}^n E_i \text{gamb}(\text{back}_{\text{opt}}(T_i)) \middle| \text{ev}(T) \right) \\
&= \text{opt} \left(\bigoplus_{i=1}^n E_i \text{gamb}(\text{norm}_{\text{opt}}(T_i)) \middle| \text{ev}(T) \right) \\
&= \text{opt} \left(\bigoplus_{i=1}^n E_i \text{opt}(\text{gamb}(T_i) | \text{ev}(T) \cap E_i) \middle| \text{ev}(T) \right),
\end{aligned}$$

whence equality follows from Lemma 48.

Finally, suppose that the root of the consistent tree T is a decision node, that is $T = \bigsqcup_{i=1}^n T_i$. We show that $\text{back}_{\text{opt}}(T) = \text{norm}_{\text{opt}}(T)$. By Lemma 50, it suffices to show that $\text{gamb}(\text{back}_{\text{opt}}(T)) = \text{gamb}(\text{norm}_{\text{opt}}(T))$. Indeed,

$$\begin{aligned}
\text{gamb}(\text{norm}_{\text{opt}}(T)) &= \text{opt}(\text{gamb}(T) | \text{ev}(T)) \\
&= \text{opt} \left(\text{gamb} \left(\bigsqcup_{i=1}^n T_i \right) \middle| \text{ev}(T) \right) \\
&= \text{opt} \left(\bigcup_{i=1}^n \text{gamb}(T_i) \middle| \text{ev}(T) \right),
\end{aligned}$$

and,

$$\begin{aligned}
\text{gamb} \left(\text{norm}_{\text{opt}} \left(\bigsqcup_{i=1}^n \text{back}_{\text{opt}}(T_i) \right) \right) &= \text{opt} \left(\text{gamb} \left(\bigsqcup_{i=1}^n \text{back}_{\text{opt}}(T_i) \right) \middle| \text{ev}(T) \right) \\
&= \text{opt} \left(\bigcup_{i=1}^n \text{gamb}(\text{back}_{\text{opt}}(T_i)) \middle| \text{ev}(T) \right) \\
&= \text{opt} \left(\bigcup_{i=1}^n \text{gamb}(\text{norm}_{\text{opt}}(T_i)) \middle| \text{ev}(T) \right) \\
&= \text{opt} \left(\bigcup_{i=1}^n \text{opt}(\text{gamb}(T_i) | \text{ev}(T_i)) \middle| \text{ev}(T) \right) \\
&= \text{opt} \left(\bigcup_{i=1}^n \text{opt}(\text{gamb}(T_i) | \text{ev}(T)) \middle| \text{ev}(T) \right),
\end{aligned}$$

whence equality follows by Lemma 45 and Property 11.

Concluding, we have shown that the implication holds for consistent decision trees consisting of a single nodes, and that if the implication holds for all children of the root node then it also holds for the whole tree. By induction, the implication holds for any consistent decision tree. \square

A.11. Proof of Lemma 26.

Proof. Assume Property 2 holds.

Consider any non-empty finite sets of normal form decisions \mathcal{X} and \mathcal{Y} , and any event $A \neq \emptyset$, such that $\text{opt}(\mathcal{X}|A) \subseteq \mathcal{Y} \subseteq \mathcal{X}$. By Property 2, it follows that $\text{opt}(\mathcal{Y}|A) = \text{opt}(\mathcal{X}|A) \cap \mathcal{Y}$, which is equal to $\text{opt}(\mathcal{X}|A)$ because $\text{opt}(\mathcal{X}|A) \subseteq \mathcal{Y}$. This proves Property 8.

Consider any non-empty finite sets of normal form decisions \mathcal{X} and \mathcal{Y} , and any event $A \neq \emptyset$ such that $\mathcal{Y} \subseteq \mathcal{X}$. If $\text{opt}(\mathcal{X}|A) \cap \mathcal{Y} = \emptyset$, then obviously $\text{opt}(\mathcal{Y}|A) \supseteq \text{opt}(\mathcal{X}|A) \cap \mathcal{Y}$. If not, then $\text{opt}(\mathcal{Y}|A) = \text{opt}(\mathcal{X}|A) \cap \mathcal{Y}$. So, Property 9 follows. \square

A.12. Proof of Theorem 29.

Lemma 54. *Let norm be any normal form operator. Let T be a consistent decision tree. If,*

(i) *for all nodes $K \in \text{ch}(T)$ such that K is in at least one element of $\text{norm}(T)$,*

$$\text{st}_K(\text{norm}(T)) \subseteq \text{norm}(\text{st}_K(T)),$$

(ii) *and, for all nodes $K \in \text{ch}(T)$, and all nodes $L \in \text{st}_K(T)$ such that L is in at least one element of $\text{norm}(\text{st}_K(T))$,*

$$\text{st}_L(\text{norm}(\text{st}_K(T))) \subseteq \text{norm}(\text{st}_L(\text{st}_K(T))),$$

then, for all nodes N in T such that N is in at least one element of $\text{norm}(T)$,

$$\text{st}_N(\text{norm}(T)) \subseteq \text{norm}(\text{st}_N(T)).$$

Proof. If N is the root of T , then the result is immediate. If $N \in \text{ch}(T)$, then the result follows from (i). Otherwise, N must be in $\text{st}_K(T)$ for one $K \in \text{ch}(T)$.

By assumption, there is a $U \in \text{norm}(T)$ that contains N (and of course also K). Therefore, $U \in \text{st}_K(\text{norm}(T))$, and by (i), $\text{st}_K(U) \in \text{norm}(\text{st}_K(T))$, and so N is also in at least one element of $\text{norm}(\text{st}_K(T))$.

We use the fact that, if \mathcal{U} and \mathcal{V} are sets of normal form decisions such that $\mathcal{U} \subseteq \mathcal{V}$, then for any node N , $\text{st}_N(\mathcal{U}) \subseteq \text{st}_N(\mathcal{V})$. Combining everything, by (i),

$$\text{st}_N(\text{st}_K(\text{norm}(T))) \subseteq \text{st}_N(\text{norm}(\text{st}_K(T)))$$

hence, since N is in at least one element of $\text{norm}(\text{st}_K(T))$, by (ii) we have

$$\subseteq \text{norm}(\text{st}_N(\text{st}_K(T))),$$

whence the desired result follows, since $\text{st}_N(\text{st}_K(T)) = \text{st}_N(T)$. \square

The following results are very similar to Lemmas 51, 52, and 53.

Lemma 55. *If norm_{opt} is subtree perfect for normal form decisions, then opt satisfies Property 7.*

Proof. Let A and B be non-empty events, and \mathcal{X}, \mathcal{Z} be non-empty finite sets of gambles, such that the following properties hold: $A \cap B \neq \emptyset$, $\bar{A} \cap B \neq \emptyset$, \mathcal{X} is $A \cap B$ -consistent, \mathcal{Z} is $\bar{A} \cap B$ -consistent, and there are $X, Y \in \mathcal{X}$ such that $AX = AY$, $X \in \text{opt}(\mathcal{X}|A \cap B)$, and $AX \oplus \bar{A}Z \in \text{opt}(\mathcal{X} \oplus \bar{A}\mathcal{Z}|B)$ for at least one $Z \in \mathcal{Z}$. If it is not possible to construct such a situation, then opt satisfies Property 7 automatically. Otherwise, to prove that Property 7 holds, we must show that $Y \in \text{opt}(\mathcal{X}|A \cap B)$.

Consider a consistent decision tree $T = AT_1 \odot \overline{A}T_2$, where $\text{ev}(T) = B$, $\text{gamb}(T_1) = \mathcal{X}$, and $\text{gamb}(T_2) = \mathcal{Z}$. Since \mathcal{X} is $A \cap B$ -consistent and \mathcal{Z} is $\overline{A} \cap B$ -consistent, we know from Definition 9 that there is such a T . By Lemma 42, there is a normal form decision $U \in \text{nfd}(T_1)$ such that $\text{gamb}(U) = \{Y\}$, and a normal form decision in $V \in \text{nfd}(T_2)$ such that $\text{gamb}(V) = \{Z\}$.

Since, by assumption, $AX = AY$, obviously $AX \oplus \overline{A}Z = AY \oplus \overline{A}Z$, and hence, also $AY \oplus \overline{A}Z \in \text{opt}(A\mathcal{X} \oplus \overline{A}\mathcal{Z}|B)$. Therefore, by definition of norm_{opt} , $AU \odot \overline{A}V \in \text{norm}_{\text{opt}}(T)$. In particular, $U \in \text{st}_N(\text{norm}_{\text{opt}}(T))$, where N is the root of T_1 . Because norm_{opt} is subtree perfect for normal form decisions, it follows that also $U \in \text{norm}_{\text{opt}}(\text{st}_N(T)) = \text{norm}_{\text{opt}}(T_1)$. Again applying the definition of norm_{opt} , we conclude that indeed $Y \in \text{opt}(\mathcal{X}|A \cap B)$. \square

Lemma 56. *If norm_{opt} is subtree perfect for normal form decisions, then opt satisfies Property 9.*

Proof. Let A be a non-empty event, and let \mathcal{X} be a non-empty finite set of A -consistent gambles. Let \mathcal{Y} be a non-empty subset of \mathcal{X} . Let $T = T_1 \sqcup T_2$, where $\text{ev}(T) = A$, $\text{gamb}(T_1) = \mathcal{Y}$, and $\text{gamb}(T_2) = \mathcal{X}$. Let N be the root of T_1 .

If $\text{opt}(\mathcal{X}|A) \cap \mathcal{Y} = \emptyset$ then Property 9 holds automatically. Suppose $\text{opt}(\mathcal{X}|A) \cap \mathcal{Y} \neq \emptyset$. By definition of norm_{opt} , N appears in at least one element of $\text{norm}_{\text{opt}}(T)$, and

$$\text{opt}(\mathcal{X}|A) \cap \mathcal{Y} = \text{gamb}(\text{st}_N(\text{norm}_{\text{opt}}(T))),$$

and by subtree perfectness of normal form decisions

$$\begin{aligned} &\subseteq \text{gamb}(\text{norm}_{\text{opt}}(T_1)) \\ &= \text{opt}(\mathcal{Y}|A). \end{aligned}$$

\square

Lemma 57. *If norm_{opt} is subtree perfect for normal form decisions, then opt satisfies Property 10.*

Proof. Let A and B be non-empty events such that $A \cap B \neq \emptyset$ and $\overline{A} \cap B \neq \emptyset$, let \mathcal{X} be a non-empty finite set of $A \cap B$ -consistent gambles, and let Z be an $\overline{A} \cap B$ -consistent gamble. Let $T = AT_1 \odot \overline{A}T_2$ be a consistent decision tree, where $\text{ev}(T) = B$, $\text{gamb}(T_1) = \mathcal{X}$, and T_2 is simply a normal form decision with $\text{gamb}(T_2) = \{Z\}$. The existence of T is assured by $A \cap B$ -consistency of \mathcal{X} and $\overline{A} \cap B$ -consistency of $\{Z\}$. Let N be the root of T_1 .

Consider any gamble $AX \oplus \overline{A}Z \in \text{opt}(A\mathcal{X} \oplus \overline{A}Z|B)$. By Lemma 42, there is a $U \in \text{nfd}(T_1)$ such that $\text{gamb}(U) = X$. By definition of norm_{opt} , it follows that $AU \odot \overline{A}T_2 \in \text{norm}_{\text{opt}}(T)$, and hence, in particular, $U \in \text{st}_N(\text{norm}_{\text{opt}}(T))$. By subtree perfectness for normal form decisions, $U \in \text{norm}_{\text{opt}}(T_1)$. Again applying the definition of norm_{opt} , we find that $X \in \text{opt}(\mathcal{X}|A \cap B)$, thus indeed $AX \oplus \overline{A}Z \in A \text{opt}(\mathcal{X}|A \cap B) \oplus \overline{A}Z$, whence Property 10 is established. \square

We can now prove Theorem 29.

Proof of Theorem 29. “only if”. Follows from Lemmas 55, 56, and 57.

“if”. We proceed as usual by structural induction. The base step is trivial as usual. Let $\text{ch}(T) = \{K_1, \dots, K_n\}$ and let $T_i = \text{st}_{K_i}(T)$. The induction hypothesis

says that norm_{opt} is subtree perfect for normal form decisions on all T_i . More precisely, for all T_i , and for every L that is in at least one element of $\text{norm}_{\text{opt}}(T_i)$,

$$\text{st}_L(\text{norm}_{\text{opt}}(T_i) \subseteq \text{norm}_{\text{opt}}(\text{st}_L(T))).$$

We must show that, for any N in at least one element of $\text{norm}_{\text{opt}}(T)$,

$$\text{st}_N(\text{norm}(T)) \subseteq \text{norm}(\text{st}_N(T)).$$

By the induction hypothesis and Lemma 54, it suffices to show this only for $N \in \text{ch}(T)$, that is, to show that

$$(26) \quad \text{st}_{K_i}(\text{norm}_{\text{opt}}(T)) \subseteq \text{norm}_{\text{opt}}(T_i)$$

for each i such that K_i is in at least one element of $\text{norm}_{\text{opt}}(T)$.

Suppose the root of T is a decision node, so $T = \bigsqcup_{i=1}^n T_i$. Let U be an element of $\text{norm}_{\text{opt}}(T)$. There is a j such that K_j is in U ; let U_j denote $\text{st}_{K_j}(U)$. To establish Eq. (26) we must show that $U_j \in \text{norm}_{\text{opt}}(T_j)$.

Note that $\text{gamb}(U_j) = \text{gamb}(U) \subseteq \text{opt}(\text{gamb}(T)|\text{ev}(T))$ since $U \in \text{norm}_{\text{opt}}(T)$. Obviously, also $\text{gamb}(U_j) \in \text{gamb}(T_j)$ by definition of gamb . Hence, it must hold that

$$\text{gamb}(U_j) \subseteq \text{opt}(\text{gamb}(T)|\text{ev}(T)) \cap \text{gamb}(T_j),$$

but, also, because $\text{gamb}(T_j) \subseteq \text{gamb}(T)$, and once noted that $\text{ev}(T) = \text{ev}(T_j)$, it follows from Property 9 that

$$\subseteq \text{opt}(\text{gamb}(T_j)|\text{ev}(T_j))$$

Putting everything together, we confirm that $U_j \in \text{norm}_{\text{opt}}(T_j)$. This proves the induction step for decision nodes.

Now suppose that the root of T is a chance node, so $T = \odot_{i=1}^n E_i T_i$. Again, let $U = \odot_{i=1}^n E_i U_i \in \text{norm}_{\text{opt}}(T)$. To establish Eq. (26) we must show that $U_i \in \text{norm}_{\text{opt}}(T_i)$ for all i .

Indeed, since $U \in \text{norm}_{\text{opt}}(T)$,

$$\text{gamb}(U) \in \text{opt}(\text{gamb}(T)) = \text{opt} \left(\bigoplus E_i \text{gamb}(T_i) \middle| \text{ev}(T) \right)$$

so by Lemma 47,

$$\subseteq \bigoplus E_i \text{opt}(\text{gamb}(T_i)|\text{ev}(T) \cap E_i).$$

So, for each T_i , there is a normal form decision $V_i \in \text{norm}_{\text{opt}}(T_i)$ such that $E_i \text{gamb}(V_i) = E_i \text{gamb}(U_i)$. Can we apply Property 7?

Obviously, $\{\text{gamb}(V_i), \text{gamb}(U_i)\} \subseteq \text{gamb}(T_i)$, and

$$E_i \text{gamb}(V_i) \oplus \overline{E_i} Z = \text{gamb}(U)$$

$$\subseteq \text{opt}(\bigoplus E_i \text{gamb}(T_i)|\text{ev}(T)) = \text{opt}(E_i \text{gamb}(T_i) \oplus \overline{E_i} Z|\text{ev}(T))$$

for suitable choices for Z and a $Z \in \mathcal{Z}$. Therefore we can apply Property 7 to conclude that U_i is in $\text{norm}_{\text{opt}}(T_i)$ for each i . This proves the induction step for chance nodes. \square

A.13. Proofs for Section 7.2.

Proof of Lemma 30. If N is in at least one element of $\text{norm}(T)$ but does not appear in $\text{ext}(T)$, then N will not appear in any element of $\text{nfd}(\text{ext}(T))$ and equivalence will fail. If a node M is in $\text{ext}(T)$ but not in any element of $\text{norm}(T)$, then M will appear in at least one element of $\text{nfd}(\text{ext}(T))$ and equivalence will fail. Therefore a node is in $\text{ext}(T)$ if and only if it is in at least one element of $\text{norm}(T)$. \square

Proof of Lemma 31. “if”. Suppose ext is subtree perfect and a node N is in $\text{ext}(T)$, then N is in at least one element of $\text{norm}(T)$. Because nfd and st_N commute, we have

$$\begin{aligned}\text{norm}(\text{st}_N(T)) &= \text{nfd}(\text{ext}(\text{st}_N(T))) = \text{nfd}(\text{st}_N(\text{ext}(T))) \\ &= \text{st}_N(\text{nfd}(\text{ext}(T))) = \text{st}_N(\text{norm}(T)).\end{aligned}$$

This demonstrates subtree perfectness of norm .

“only if”. By Lemma 30, for a particular norm there can be no more than one equivalent ext . We show that this ext is subtree perfect. A node N is in this ext if and only if it is in at least one element of $\text{norm}(T)$. Similarly, a node M is in $\text{st}_N(\text{ext}(T))$ if and only if M is in at least one element of $\text{st}_N(\text{norm}(T))$. By subtree perfectness of norm , the latter is satisfied if and only if M is in at least one element of $\text{norm}(\text{st}_N(T))$. But, again by definition of ext , the latter is satisfied if and only if M is in $\text{ext}(\text{st}_N(T))$. This establishes subtree perfectness of ext . \square

Proof of Theorem 32. By Lemma 31, if an equivalent ext exists then it is subtree perfect. We must show that the ext constructed in Lemma 30 satisfies

$$\text{norm}_{\text{opt}}(T) = \text{nfd}(\text{ext}(T))$$

for all consistent decision trees T .

We now proceed by structural induction. The base step, that $\text{nfd}(\text{ext}(T)) = \text{norm}_{\text{opt}}(T)$ for any decision tree comprising only a single node, is as usual satisfied trivially.

Let us proceed with the induction step. The induction hypothesis states that, for any node K in $\text{ch}(T)$, $\text{nfd}(\text{ext}(\text{st}_K(T))) = \text{norm}_{\text{opt}}(\text{st}_K(T))$. We must show that $\text{norm}_{\text{opt}}(T) = \text{nfd}(\text{ext}(T))$.

It is useful to show first that, if \mathcal{K} is the set of all $K \in \text{ch}(T)$ that appear in at least one element of $\text{norm}_{\text{opt}}(T)$ (or equivalently, that appear in $\text{ext}(T)$), then for any $K \in \mathcal{K}$,

$$(27) \quad \text{st}_K(\text{nfd}(\text{ext}(T))) = \text{st}_K(\text{norm}_{\text{opt}}(T)).$$

Consider any node K in $\text{ch}(T)$ that appears in $\text{ext}(T)$. Clearly,

$$\text{st}_K(\text{nfd}(\text{ext}(T))) = \text{nfd}(\text{st}_K(\text{ext}(T)))$$

and since we just proved that ext is subtree perfect,

$$= \text{nfd}(\text{ext}(\text{st}_K(T)))$$

and by the induction hypothesis,

$$= \text{norm}_{\text{opt}}(\text{st}_K(T))$$

but, by definition of ext , the node K also appears in at least one element of $\text{norm}_{\text{opt}}(T)$, so by the subtree perfectness of norm_{opt} ,

$$= \text{st}_K(\text{norm}_{\text{opt}}(T)).$$

This establishes Eq. (27).

Now, suppose that the root of T is a decision node. Observe that

$$\text{nfd}(\text{ext}(T)) = \bigsqcup_{K \in \mathcal{K}} \text{nfd}(\text{st}_K(\text{ext}(T))),$$

and since $\text{st}(\cdot)$ and $\text{nfd}(\cdot)$ commute,

$$= \bigsqcup_{K \in \mathcal{K}} \text{st}_K(\text{nfd}(\text{ext}(T))).$$

Also, because opt satisfies Properties 1 and 2, we have (as seen in the proof of Theorem 22)

$$\text{norm}_{\text{opt}}(T) = \bigsqcup_{K \in \mathcal{K}} \text{st}_K(\text{norm}_{\text{opt}}(T)),$$

whence by Eq. (27),

$$\text{norm}_{\text{opt}}(T) = \text{nfd}(\text{ext}(T)).$$

Finally, suppose that the root of T is a chance node. Here, \mathcal{K} is simply $\text{ch}(T) = \{K_1, \dots, K_n\}$. Similarly to before, we have

$$\begin{aligned} \text{nfd}(\text{ext}(T)) &= \bigcirc_{i=1}^n E_i \text{nfd}(\text{st}_{K_i}(\text{ext}(T))) \\ &= \bigcirc_{i=1}^n E_i \text{st}_{K_i}(\text{nfd}(\text{ext}(T))). \end{aligned}$$

Since opt satisfies Properties 1, 2, and 3, we have (as seen in the proof of Theorem 22),

$$\begin{aligned} \text{norm}_{\text{opt}}(T) &= \bigcirc_{i=1}^n E_i \text{norm}_{\text{opt}}(\text{st}_{K_i}(T)) \\ &= \bigcirc_{i=1}^n E_i \text{st}_{K_i}(\text{norm}_{\text{opt}}(T)), \end{aligned}$$

whence by Eq. (27), we have

$$\text{nfd}(\text{ext}(T)) = \text{norm}_{\text{opt}}(T).$$

□

E-mail address: `nath.huntley@gmail.com`

DURHAM UNIVERSITY, DEPARTMENT OF MATHEMATICAL SCIENCES, SCIENCE LABORATORIES,
SOUTH ROAD, DURHAM DH1 3LE, UNITED KINGDOM

E-mail address: `matthias.troffaes@durham.ac.uk`

DURHAM UNIVERSITY, DEPARTMENT OF MATHEMATICAL SCIENCES, SCIENCE LABORATORIES,
SOUTH ROAD, DURHAM DH1 3LE, UNITED KINGDOM